Private Training & Data Generation by Clustering Embeddings

Felix Zhou
Yale University
felix.zhou@yale.edu

Samson Zhou
Texas A&M University
samsonzhou@gmail.com

Vahab Mirrokni
Google Research
mirrokni@google.com

Alessandro Epasto
Google Research
aepasto@google.com

Vincent Cohen-Addad
Google Research
cohenaddad@google.com

Abstract

Deep neural networks often use large, high-quality datasets to achieve high performance on many machine learning tasks. When training involves potentially sensitive data, this process can raise privacy concerns, as large models have been shown to unintentionally memorize and reveal sensitive information, including reconstructing entire training samples. Differential privacy (DP) provides a robust framework for protecting individual data and in particular, a new approach to privately training deep neural networks is to approximate the input dataset with a privately generated synthetic dataset, before any subsequent training algorithm. We introduce a novel principled method for DP synthetic image embedding generation, based on fitting a Gaussian Mixture Model (GMM) in an appropriate embedding space using DP clustering. Our method provably learns a GMM under separation conditions. Empirically, a simple two-layer neural network trained on synthetically generated embeddings achieves state-of-the-art (SOTA) classification accuracy on standard benchmark datasets. Additionally, we demonstrate that our method can generate realistic synthetic images that achieve downstream classification accuracy comparable to SOTA methods. Our method is quite general, as the encoder and decoder modules can be freely substituted to suit different tasks. It is also highly scalable, consisting only of subroutines that scale linearly with the number of samples and/or can be implemented efficiently in distributed systems.

1 Introduction

The rise of massive datasets and increasingly complex machine learning (ML) models has transformed a large number of fields such as computer vision, natural language processing, and pattern recognition. These advancements have been fueled by the availability of high-quality datasets, enabling deep neural networks to achieve unprecedented performance across diverse tasks. However, the widespread reliance on large-scale data in ML introduces significant challenges and potential risks. One such risk is inadvertently exposing private user information in the output of a machine learning system [SM21]. These risks have led to the establishment of strict data privacy regulations that forbid the storage of data that can be re-traced to individuals (re-identification) [VV17]. Thus, privacy-preserving

Fig. 1: Synthetic and original CIFAR-10 images at $\varepsilon = 8$, $\delta = 10^{-5}$. Each row corresponds to a different class. The left-most columns are synthetic images obtained with our method, while the right-most columns are original images.

machine learning is no longer only a desirable property, but a necessity. When dealing with private data, differential privacy (DP) [DMN+06] (Definition D.3) has emerged as the gold standard for ensuring strong privacy protection. DP ensures that outputs of an algorithm are statistically similar regardless of the inclusion of any single data point, thus provably avoiding privacy risks such as re-identification. As such, DP presents a strong framework for regulation-compliant training on sensitive data [CD18].

In this paper, we study the problem of differentially-private synthetic data generation [AZK+18; TKP19; RLP+20; MJW+22; TFR22; GBG+23; HJS+23; KPS+23; YIL+23; HSZ+24; LGK+24; XLB+24; AAB+25; TXX+25]. The goal of DP synthetic data release is to privately obtain approximations of potentially sensitive datasets that effectively extract, from the data, the useful information needed to achieve the system's goals, while at the same time ensuring that no individual's privacy is compromised.

Specifically, consider the problem of training a machine learning model for a certain task (e.g., classification) with DP guarantees [ACG+16]. In this context, DP synthetic data can be used to output a privatized version of the training dataset (see Figure 1), where then arbitrary *non*-private training techniques can be applied without additional privacy risk. This approach is an increasingly popular alternative to directly training an ML model for the task using DP-SGD [ACG+16; DBH+22; YNB+22; HLY+23; MGN+23] due to several advantages over private model training:

(1) Publishing synthetic datasets can enable direct inspection of an approximation of the underlying data, allowing model designers the freedom to explore the data to identify issues, debug model behaviors, and assess data quality.

(2) DP synthetic data generation allows plug-in use of any existing model architecture without the need to run more complex privacy-preserving training methods, such as Differentially Private Stochastic Gradient Descent (DP-SGD) [ACG+16; DBH+22]. This avoids the additional

engineering effort needed to support DP training pipelines, which may require a fine-grained understanding of the interplay between privacy and the underlying training mechanics. For instance, Opacus [YSS+21], a popular DP training library, requires a custom implementation of a per-sample gradient calculator for custom layers.

(3) Private synthetic data release allows for unlimited training of models without incurring additional privacy costs. By comparison, repeated private training requires accounting for accumulated privacy loss.

In this work, we design novel methods for practical DP synthetic data generation by taking inspiration from the embedding clustering literature [XGF16]. Our guiding insight is that an appropriate embedding of the input data makes it more amenable to clustering. Indeed, embedding objects into an appropriate space and clustering these embeddings has been theoretically [Lux07; Spi25] and empirically [HHW+14; JZT+17; RSB+19; RDS+19] shown to be effective at capturing desirable structures within data.

1.1 Problem Definition

Our overarching goal is to develop private synthetic data to perform downstream tasks. Though our methodologies focus on image classification, we first provide a formal model to quantify the performance of a synthetic dataset for general classification. Formally, consider an ML task where the goal is to perform classification on an input space $\mathcal{X}$ for a label space $\mathcal{Y} = \{1, \dots, L\}$. We remark that $\mathcal{X}$ can either be the original input dataset or an embedding of the dataset under any fixed encoding scheme.

Suppose the loss function $\ell(\cdot, \cdot; \theta) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is parameterized by the vector θ over the hypothesis class. Here, the vector θ denotes the parameters of the deep learning model, e.g., the weights, biases, and hyperparameters of the model. We assume access to a collection S of n data points that are sampled i.i.d. over the space $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ so that $(x_i, y_i) \sim \mathcal{D}_{\mathcal{Z}}$ for each $i \in [n]$, for some probability distribution $\mathcal{D}_{\mathcal{Z}}$. We would like to apply a learning algorithm $\mathcal{A}$ onto the input S to learn a model that can accurately predict the correct labels for new, unseen data by capturing the underlying patterns or relationships in the training data, while simultaneously protecting potentially sensitive information. Our approach is to privately estimate the true distribution, say with $\widetilde{\mathcal{D}}_{\mathcal{Z}}$, and release a private synthetic dataset $\widetilde{S}$ from $\widetilde{\mathcal{D}}_{\mathcal{Z}}$, such that any algorithm $\mathcal{A}$ trained on $\widetilde{S}$ retains good accuracy when applied to $\mathcal{D}$ rather than $\widetilde{S}$. Quantitatively, our goal is to minimize the classification loss of the private synthetic dataset, which can be decomposed as follows:

$$\begin{aligned}\mathop{\mathbb{E}}_{\mathbf{x},y\sim\mathcal{D}_{\mathcal{Z}}}[\ell(\mathbf{x}, y; \theta)] \leq \underbrace{\frac{1}{|\widetilde{S}|}\sum_{\tilde{\mathbf{x}},\tilde{y}\in\widetilde{S}} \ell(\tilde{\mathbf{x}}, \tilde{y}; \theta)}_{\text{training error}} + \underbrace{\left|\frac{1}{|\widetilde{S}|}\sum_{\tilde{\mathbf{x}},\tilde{y}\in\widetilde{S}} \ell(\tilde{\mathbf{x}}, \tilde{y}; \theta) - \mathop{\mathbb{E}}_{\tilde{\mathbf{x}},\tilde{y}\sim\widetilde{\mathcal{D}}_{\mathcal{Z}}}[\ell(\tilde{\mathbf{x}}, \tilde{y}; \theta)]\right|}_{\text{synthetic data generation error}} \\ + \underbrace{\left|\mathop{\mathbb{E}}_{\tilde{\mathbf{x}},\tilde{y}\sim\widetilde{\mathcal{D}}_{\mathcal{Z}}}[\ell(\tilde{\mathbf{x}}, \tilde{y}; \theta)] - \mathop{\mathbb{E}}_{\mathbf{x},y\sim\mathcal{D}_{\mathcal{Z}}}[\ell(\mathbf{x}, y; \theta)]\right|}_{\text{estimation error}},\end{aligned} \tag{1}$$

where $\widetilde{S} = \{\widetilde{\mathbf{x}_j}, \widetilde{y_j}\}$ is a (private) synthetic dataset.

Thanks to the postprocessing property of DP, we can sample as many points from our privately estimated distribution $\widetilde{\mathcal{D}}_{\mathcal{Z}}$ as desired. Hence we do not focus on the generation error but note that

under mild assumptions, we can show that the generation error converges uniformly to 0 across all parameters $\theta \in \Theta$ using techniques such as metric entropy [Wai19]. Thus our primary concern is to develop a principled DP distribution estimation algorithm.

Loss function regularity conditions. We remark that although the problem formulation is simple, there is no upper bound to the private synthetic data loss without additional assumptions on the loss function. To circumvent these limitations due to poorly behaved loss functions, existing works often assume the dataset lies in a metric space that is "well-behaved" with respect to the loss function of the model. For example, Sener and Savarese [SS18] assumes the loss function is λ-Lipschitz, i.e., $|\ell(\mathbf{x}, y) - \ell(\mathbf{x}', y)| \leq \lambda \cdot \|\mathbf{x} - \mathbf{x}'\|_2$, while Axiotis, Cohen-Addad, Henzinger, et al. [ACH+24] assumes the loss function is (z, λ)-Hölder continuous, i.e., $|\ell(\mathbf{x}, y) - \ell(\mathbf{x}', y)| \leq \lambda \cdot \|\mathbf{x} - \mathbf{x}'\|_2^z$ and demonstrate experimentally that this holds true for in the context of large language models (T5-model [RSR+20] for a translation task and BERT embeddings [DCL+19a]).

Embedding space. Functionally, the elements of the dataset can be embedded into a metric space, e.g., graph embeddings [GL16], word embeddings [MCC+13; PSM14; DCL+19b], or image embeddings [SZ15; HZR+16; RKH+21]. In general, an embedding can be acquired from the last layers of a neural network, which is especially appropriate when the model has already been pre-trained on publicly available data and the goal is to either fine-tune the model on private data for a specific task. In these settings, a natural view is that the input dataset to the algorithm is the embedding of the original dataset, while the loss function may be the norm of the gradient of the embedding.

1.2 Our Contributions

In this paper, we present a novel training-free approach based on Gaussian mixture models (GMMs, c.f., Definition D.1) to privately generate synthetic data to minimize the error specified in Equation (1) after training. We first seek to privately partition the input dataset into k clusters, adapting a recent line of work [SS18; ACH+24] for the non-private active learning problem. Existing works use the resulting clustering to sample a number of points from each cluster, a procedure that inherently violates differential privacy. Instead, we privately estimate the intra-cluster covariance as our goal is to release a private synthetic dataset based on the resulting clustering. Informally, we would like to preserve the distribution of the input points, since the sample distribution serves as an estimate of the true distribution. The main intuition is that if the dataset can be partitioned into k clusters such that each cluster can be well-approximated by a Gaussian distribution, then by generating data points using a GMM, we expect that the distributional distance between generated points and the input distribution to be small.

Theorem 1.1 (Informal Parameter Estimation; See Theorem G.7)**.** *Let $\varepsilon, \delta, \alpha, \beta \in (0, 1)$. Given n samples from a well-separated k-Gaussian mixture model $\mathcal{D}_{\text{GMM}}$ in d-dimensional space for $n = \text{poly}(k, d, 1/\alpha, 1/\varepsilon, \log(1/\beta), \log(1/\delta))$, Algorithm 1 is an (ε, δ)-DP algorithm that outputs parameter estimates $\hat{w}_i, \hat{\mu}_i, \hat{\Sigma}_i$ such that with probability $1 - \beta$, $\|w - \hat{w}\|_1, \|\mu - \hat{\mu}\|_2, \left\|\Sigma - \hat{\Sigma}\right\|_F \leq \alpha$.*

Algorithms for probably learning GMMs have been well-studied by the Theoretical Computer Science community. Our algorithm accomplishing Theorem 1.1 follows the well-studied cluster-then-learn paradigm [Das99], which requires some form of separation condition of the underlying distribution. See Section 1.3 for more details.

Now, for a "well-behaved" loss function, e.g., Lipschitz, we can provably approximate the loss of the original dataset for the purposes of downstream training. Moreover, our parameter estimation algorithm yields a conditional generation algorithm for labeled data by running the estimation algorithm for each class.

Theorem 1.2 (Informal Downstream Training; See Theorem G.11)**.** *Let $\varepsilon, \delta, \alpha, \beta \in (0,1)$ and f be a (λ, z)-Hölder continuous loss function for $z \in [1,2]$. Suppose $Z = (X, Y)$ is a joint feature-label distribution for $Y \in [c]$ where each conditional distribution $(X \mid Y = y) \sim \mathcal{D}^{(y)}_{\mathrm{GMM}}$ is a well-separated Gaussian mixture model. Given n samples from each conditional distribution for $n = \mathrm{poly}(k, d, 1/\alpha, 1/\varepsilon, \log(1/\beta), \log(1/\delta))$, there is an (ε, δ)-DP algorithm that outputs a distribution $\tilde{Z} = (\tilde{X}, Y)$ such that with probability $1 - \beta$, $\mathop{\mathbb{E}}_{Z}[f(Z)] \leq \mathop{\mathbb{E}}_{\tilde{Z}}\left[f(\tilde{Z})\right] + \lambda \cdot \alpha$.*

We also show that our algorithm satisfies (ε, δ)-DP and can be implemented in near-linear time.

Theorem 1.3 (Informal; See Theorems G.1 and G.2)**.** *Let $(\varepsilon, \delta) \in (0,1)$, n be the number of input images, T be the maximum runtime of* `Encode` *and* `Decode` *on a single input, and d the embedding dimension. Algorithm 1 is (ε, δ)-DP and can be implemented in $\tilde{O}(n(d+T) \cdot \mathrm{poly}(1/\varepsilon, \log(1/\delta)))$ time.*

We implement and test our framework on standard benchmark datasets from DP classification and synthetic data literature [TKP19; DBH+22; GBG+23; LGK+24] at the same privacy levels as the state-of-the-art (SOTA) DP classification [DBH+22]. While our theoretical analysis hinges on separability conditions, we find that our method empirically yields strong downstream classification accuracy regardless. Specifically, we train a simple two-layer neural network on DP synthetic embeddings and compare its accuracy against all DP training methods, including those that do not use synthetic data. We obtain SOTA classification accuracy on standard datasets in the DP synthetic data literature (See Section 3).

Note that one would expect training via DP synthetic data generation to achieve worse performance than direct training via DP-SGD, as the former is a more general task. This belief is supported by previous work on DP synthetic image generation [LGK+24]. Thus it is very surprising that we can achieve comparable, not to mention new SOTA DP classification results.

1.3 Related Works

There are many related works that are relevant to this paper. We discuss the immediately related works and defer the rest to Appendix B.

DP synthetic data. Given a private dataset D, the goal is to privately generate a synthetic dataset which is statistically similar to D [AZK+18; TKP19; RLP+20; MJW+22; TFR22; GBG+23; HJS+23; KPS+23; YIL+23; HSZ+24; LGK+24; XLB+24]. See [CKF24; HWL+24] and references therein for a survey of recent developments. One related line of work on DP synthetic data given only *API-access* to foundation models [LGK+24; XLB+24] also develops training-free methods that leverage pre-trained embeddings. However, they only do so in the context of establishing a measure of difference between a candidate synthetic dataset and the true sensitive dataset. We further leverage the power of pre-trained embeddings by clustering together similar data points in the embedding space and modeling each cluster using a Gaussian distribution.

DP clustering. DP k-means clustering seeks to identify groups of similar data points while ensuring the output is not overly sensitive to the value of any particular entry. [SCL+16; SCL+17; HL18; LWG+19]. A particularly relevant line of work is that on scalable DP clustering algorithms which terminate in near-linear running time [CEL+22; CEM+22].

(DP) Gaussian mixture models. Mixture models were introduced by Pearson [Pea94] for modeling the presence of subpopulations. The most popular algorithm for estimating GMMs in practice is a heuristic called Expectation-Maximization (EM) [DLR77]. Unfortunately, EM does not provably learn GMMs. In a seminal paper, Dasgupta [Das99] designed the first (efficient) clustering-based algorithm that provably learns a GMM under separation conditions similar to ours. The cluster-then-learn scheme introduced by Dasgupta [Das99] led to follow-up works [DS00; VW04; AK05] following said scheme that shaved the degree of separation needed. Departing from clustering-based techniques, Kalai, Moitra, and Valiant [KMV10] and Moitra and Valiant [MV10] developed sophisticated algorithms for learning GMMs without any separation conditions. Unlike clustering-based algorithms, These algorithms have polynomial dependence on relevant parameters except k (the number of components), which is unfortunately necessary in the absence of separation conditions. See e.g. [Moi18] for a more detailed survey of prior algorithmic developments.

In general, the covariance matrices within each component of a GMM (Definition D.1) can be arbitrary. However, various restrictions of the covariance structure have been studied and applied across various fields, including spherical covariances [HK13], diagonal covariances [Rey+09], and tied covariances [GRG06].

In the differential privacy community, prior works have studied the task of privately learning a GMM under various assumptions [PFC+17; KSS+19]. See Arbas, Ashtiani, and Liaw [AAL23] and references therein for a more comprehensive history. However, practical implementations have so far been underexplored. Our simple algorithm for privately fitting a GMM based on the more well-studied task of DP clustering may be of interest beyond synthetic data generation.

1.4 Preliminaries

We defer standard preliminaries to Appendix D.

Notation. We write d to denote the ambient dimension, ε, δ to denote the approximate-DP parameters, and α, β to denote the accuracy and failure probability parameters. We use k to denote the number of clusters or components for k-means or Gaussian mixture models, respectively. Typically, we use μ, Σ to denote the mean and covariance of a distribution.

2 Overview of Techniques & Utility Analysis

In this section, we provide the theoretical guarantees for our training-free pipeline. We describe our procedures and the corresponding analysis for *unconditional* generation. That is, there is no notion of a label for the dataset or equivalently, all the labels of the dataset are assumed to be the same. We remark that this is without loss of generality because for the case of *conditional* generation, it suffices to repeat the procedure and analysis in parallel for each class in the training set. We provide pseudocode for our method in Algorithm 1.

Algorithm 1 DP Synthetic Generation

Input: data $D = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, privacy parameters ε, δ, number of clusters k, number of generations m

$D_{\texttt{Embedding}} \leftarrow \{\texttt{Encode}(\mathbf{x}) : \mathbf{x} \in D\}$

$(\mathbf{c}_1, n_1), \ldots, (\mathbf{c}_k, n_k) \leftarrow \texttt{DP-Cluster}(D_{\texttt{Embedding}}, \varepsilon/5, \delta/5)$

for $j = 1, \ldots, k$ **do**

  $D_j \leftarrow \{\mathbf{x} \in D_{\texttt{Embedding}} : \mathbf{c}_j = \operatorname{argmin}_{\mathbf{c}=\mathbf{c}_1,\ldots,\mathbf{c}_k} \|\mathbf{x} - \mathbf{c}\|_2\}$

  $\mu_j \leftarrow \texttt{DP-Mean}(D_j, \varepsilon/5, \delta/5)$

  $\Sigma_j \leftarrow \texttt{DP-Covariance}(D_j, \varepsilon/5, \delta/5)$

  $p_j \leftarrow n_j / \sum_{j=1}^{k} n_j$

$Z_{\texttt{Embedding}} \leftarrow \varnothing$

for $\ell = 1, \ldots, m$ **do**

  $j \sim [k]$ with probability p_j

  $\mathbf{z}_\ell \sim \mathcal{N}(\mu_j, \Sigma_j)$

  $Z_{\texttt{Embedding}} \leftarrow Z_{\texttt{Embedding}} \cup \{z_\ell\}$

$Z \leftarrow \texttt{DP-FilterEmbedding}(Z_{\texttt{Embedding}}, D, \varepsilon/5, \delta/5)$

yield Z

$Z_{\texttt{Image}} \leftarrow \{\texttt{Decode}(\mathbf{z}) : \mathbf{z} \in Z\}$

yield $\texttt{DP-FilterImage}(Z_{\texttt{Image}}, D, \varepsilon/5, \delta/5)$

2.1 Subroutines

Encoders & decoders. Our utility analysis relies on the loss function being Hölder continuous over the input space. embedding space. While this may seem to be a strong assumption, it has been experimentally verified to hold for certain embeddings [DCL+19a; RSR+20].

Thus, to privately train a classifier by training on DP synthetic embeddings, we assume there is a publicly available encoder module `Encode` that takes a $(C \times W \times H)$ image and outputs a vector $x \in \mathbb{R}^d$. Here C is the number of image channels and W, H are the width and height of the input image.

If in addition, we wish to generate DP synthetic images, we assume access to a decoder module `Decode` that takes a vector $x \in \mathbb{R}$ and maps it back to an image, possibly of different dimensions $(C' \times W' \times H')$.

Filtering embeddings & images. Similar to any (not necessarily private) data generation process, our method may occasionally generate an embedding or an image that is a poor representation of the underlying sensitive data. Thus, our algorithm optionally supports filtering at the embedding and image level, where we discard some of the generated embeddings or images based on some rules `DP-FilterEmbedding`, `DP-FilterImage`. Similar to Hou, Shrivastava, Zhan, et al. [HSZ+24], Lin, Gopi, Kulkarni, et al. [LGK+24], and Xie, Lin, Backurs, et al. [XLB+24], we allow the filtering to depend on private data.

2.2 Synthetic Embeddings

Our full algorithm for generating synthetic data is presented in Algorithm 1. While the pseudocode includes the optional image generation step, it suffices to stop before the decoding step for the purpose of training a classifier on DP synthetic embeddings. The rest of this section delves into some details and analysis of Algorithm 1.

Encoding images. We use a variant of the pre-trained CLIP [RKH+21] image encoder to encode each training and test image into 768-dimensional embeddings. In particular, we use `CLIPImageProcessor`[1] and `CLIPVisionModelWithProjection`[2]. Both the models and model weights[3] are publicly available on HuggingFace. Note that there are no private operations in this step.

Privately learning a GMM. Next, we privately fit a k-Gaussian Mixture Model (k-GMM) on the embeddings produced by the previous step. This comprises of two steps: learning a partition of the dataset using a private k-means algorithm and privately estimating the intra-cluster means/covariances given these private centers. We analyze both steps in Appendix G.3.

Intuitively, assuming the data embeddings were generated from a k-GMM, a reasonable approximate k-means solution must place a center close to each cluster. Then, assuming the components are sufficiently well-separated, it should be the case that each output center is also well-separated and hence we can "classify" points by nearest center. We formalize this intuition in Theorem G.6.

Then, these k centers induce a partition of the dataset, where a point belongs to the i-th partition if its closest center is the i-th center. Assuming we managed to capture only points from the i-th component in the i-th partition, we can estimate the parameters of the i-th component using any algorithm for Gaussian estimation. This is made formal in Theorem G.7.

In our experiments, we use the practical DP k-means algorithm by Chang and Kamath [CK21] to privately compute k centers. Note that the number of clusters k is a tuned hyperparameter. We also output a noisy count of the number of elements within each partition (cf. Algorithm 1).

For the second step, we estimate the intra-cluster means and covariances by clipping and adding appropriate Gaussian noise. There are many variations of restricted covariance models within GMMs (see Section 1.3) and we empirically noticed that diagonal covariances yield the best performance.

Private synthetic embedding generation. Given the private k-GMM, we can then generate an unlimited number of synthetic embeddings simply by sampling from the GMM. This does not incur additional privacy loss as it is post-processing.

Optionally, we prune the generated point using noisy votes from original training data, similar to a single iteration of Private Evolution [LGK+24; XLB+24]. That is, each original embedding point votes for its nearest neighbor in the generated embeddings. After adding an appropriate amount of noise to the count to preserve privacy, we keep a generated embedding only if its noisy vote is above a certain threshold. This threshold is a hyperparameter.

[1] https://huggingface.co/docs/transformers/v4.48.0/en/model_doc/clip#transformers.CLIPImageProcessor

[2] https://huggingface.co/docs/transformers/v4.48.0/en/model_doc/clip#transformers.CLIPVisionModelWithProjection

[3] https://huggingface.co/diffusers/stable-diffusion-2-1-unclip-i2i-l/tree/main

Training a classifier on synthetic embeddings. Given a dataset of synthetic embeddings, our goal is to train a model by minimizing an appropriate well-behaved loss function over the synthetic embeddings. We analyze this step in Appendix G.4.

As mentioned before, since we can generate as many synthetic embeddings as we want, Equation (1) shows that the proxy error arising from training on synthetic embeddings should be dominated by the estimation error. We translate the parameter estimation error to a distributional bound in Wasserstein distance between GMMs, which implies a bound on the proxy error for Hölder continuous functions. This is quantified in Theorem G.11.

Experimentally, we train a simple two-layer neural network on the synthetic embeddings and test its accuracy on the *original* test set embeddings. As remarked earlier, using non-private training techniques does not incur any private loss, as the synthetic embedding generation process is differentially private. We achieve SOTA accuracy on CIFAR-10 [Ale09] and CAMELYON17 [BGM+19]. We also achieve comparable accuracy on the more challenging CIFAR-100 [Ale09] dataset. See Section 3.2 for more details.

2.3 Synthetic Images

The above already suffices to train a private classifier. If we wish to also generate images, it can be obtained with the help of a decoder module.

Decoding Embeddings into Images. We use StableUnCLIP, a stable diffusion model fine-tuned on CLIP embeddings [RBL+22] to decode CLIP embeddings into 768×768 images. Specifically, we use the class `StableUnCLIPImg2ImgPipeline`[4] with publicly available weights[5] through HuggingFace.

Optionally, we use NIQE [MSB13] and PIQE [She05; VPB+15] image quality filters to filter out the generated images that are too noisy. Note that the two pruning strategies do not depend on the private data and simply compute a "quality" score given an input image.

Training a Classifier on Synthetic Images. We use the decoded images to fine-tune a torchvision [MC16] implementation of the ResNet50 model [HZR+16] that was pre-trained on Imagenet [DDS+09]. The only modification is to change the final classification layer to match the number of classes for the dataset in question. Again, we note that any non-private training method can be used to obtain a private classifier since the training data is guaranteed to be differentially private.

3 Experiments

We begin by describing our experimental setup in Section 3.1.

In Section 3.2, we compare the classification accuracy of a simple two-layer neural network trained on DP synthetic embeddings against SOTA private training methods on standard benchmark datasets. We emphasize that we compare against all DP training methods, including those that do not utilize synthetic data like DP-SGD. Surprisingly, we achieve new SOTA results on CIFAR-10

[4] `https://huggingface.co/docs/diffusers/en/api/pipelines/stable_unclip#diffusers.StableUnCLIPImg2ImgPipeline`

[5] `https://huggingface.co/diffusers/stable-diffusion-2-1-unclip-i2i-l/tree/main`

and CAMELYON17 while obtaining competitive accuracy on the more challenging CIFAR-100 dataset.

In Section 3.3, we demonstrate that our method is also able to generate useful private synthetic images and compare the downstream classification accuracy of models trained on such images against SOTA private synthetic image generation methods for CIFAR-10. In particular, our method achieves superior classification accuracy at all privacy budgets.

Finally, Section 3.4 presents a detailed comparison of our method against DP-SGD on various privacy budgets on CIFAR-10.

3.1 Experimental Setup

Public data. The CLIP embedding module [RKH+21] was pre-trained on unspecified image-text pairs scoured from the internet. We emphasize that our experiments on synthetic embeddings only use CLIP as public data. Our results on synthetic images also requires a decoder. Our decoder module is based on Stable Diffusion 2.1 [RBL+22], which is trained on the LAION-5B [SBV+22] dataset and fine-tuned to invert CLIP embeddings using the same dataset. For classification on synthetic images, we fine-tune a model that was pre-trained on ImageNet [DDS+09]. We consider the above as publicly available data.

We selected CLIP embeddings because it is a crucial component of the only known general large pre-trained encoder-decoder model pair. We were unable to find other encoder-decoder pairs that generalize beyond the specific data sets upon which they were pre-trained.

Sensitive data. We execute our DP synthetic generation pipeline on CIFAR-10, CIFAR-100 [Ale09], and CAMELYON17 [BGM+19], which we consider as private sensitive data. The first two consist of natural images with 10 and 100 classes respectively, and the CAMELYON17 is a medical dataset for binary classification of breast cancer metastases. We emphasize that these are standard benchmark datasets within the DP synthetic images literature [TKP19; GBG+23; LGK+24].

Hyperparameter tuning. We performed hyperparameter search on the number of clusters as well as the clipping radii for the generation algorithm and followed the TorchVision formula for training[6] without hyperparameter search. Similar to other works on DP synthetic data [GBG+23; LGK+24], we do not account for hyperparameter search as part of the privacy budget.

Setup. Each training experiment is repeated for 3 runs and we report the mean accuracy and standard deviation. Our experiments are performed using eight H100 GPUs (80GB memory each). See Appendix C for more setup details for our experiments.

3.2 Private Training

We compare the downstream classification of a simple two-layer neural network trained on private synthetic embeddings against the classification accuracy of all other private training methods. See Appendix C.3 for details of the two-layer neural network.

DP-finetuning [DBH+22] achieves the current SOTA on CIFAR-10, CIFAR-100 and DP-Diffusion [GBG+23] achieves the current SOTA on CAMELYON17. We also display the non-DP

[6] https://pytorch.org/blog/how-to-train-state-of-the-art-models-using-torchvision-latest-primitives/

Table 1: Private training classification accuracies for various data sets. DP-SGD results taken from respective papers. $\delta = 10^{-5}$ for all experiments.

Dataset	ε	Ours	SOTA (DP)	SOTA (non-DP)
CIFAR-10	8	97.0 ± 0.01	96.6	99.5
CIFAR-100	8	80.5 ± 0.177	81.8	96.1
CAMELYON17	10	93.1 ± 0.067	91.1	95.7

SOTA results, as reported by Lee, Oh, Choi, et al. [LOC+19], Dosovitskiy, Beyer, Kolesnikov, et al. [DBK+21], and Foret, Kleiner, Mobahi, et al. [FKM+21]. For the above datasets, we generate the same number of synthetic embeddings as the original training splits and train a simple two-layer neural network from scratch on said embeddings. We then test on embeddings of the *original* (non-synthetic) test set. Table 1 shows that our method achieves an improvement on the SOTA for CIFAR-10 and CAMELYON17 at the same privacy budgets as prior works.

We emphasize that this runs contrary to conventional beliefs [LGK+24], as DP synthetic data is more general-purpose than training via DP-SGD, which is optimizing for a single task.

Our results suggest that even on private datasets with significant distribution shift from the encoder training data, training on synthetic embeddings can yield classifiers with strong privacy-to-utility tradeoffs.

3.3 Private Synthetic Images

Next, we compare the downstream classification accuracy of classifiers trained on synthetic images generated by decoding embeddings against other baseline DP synthetic image techniques on CIFAR-10. Private Evolution [LGK+24] achieves the current SOTA on lower values of ε while DP-Diffusion [GBG+23] achives the current SOTA on higher values of ε.

We fine-tune a ResNet50 [HZR+16] classifier pre-trained on ImageNet [DDS+09] using 50,000 DP synthetic images and test its accuracy on the original (non-synthetic) CIFAR-10 test set. Figure 2 compares the results across various levels of ε. We note that Harder, Jalali, Sutherland, et al. [HJS+23] achieve an accuracy of 51% (not shown).

The above shows that when the decoder module is pre-trained on similar data to the private dataset, our method can achieve strong utility at lower privacy budgets.

3.4 Privacy-Utility Tradeoffs

Finally, we consider the privacy-utility tradeoffs of our methods by examining the performance at varying levels of ε.

Our first comparison is quantitative, and we examine the downstream classification accuracy for both synthetic embeddings and actual images for the CIFAR-10 dataset. For reference, we consider DP-finetuning [DBH+22]. Table 2 shows that our classifier trained on synthetic embeddings consistently outperforms the one trained by DP-finetuning at various levels of ε.

Next, we qualitatively compare of the images generated at different privacy levels. See Appendix A for examples of synthetic CIFAR-10 images at various levels of ε. Interestingly, while the classification

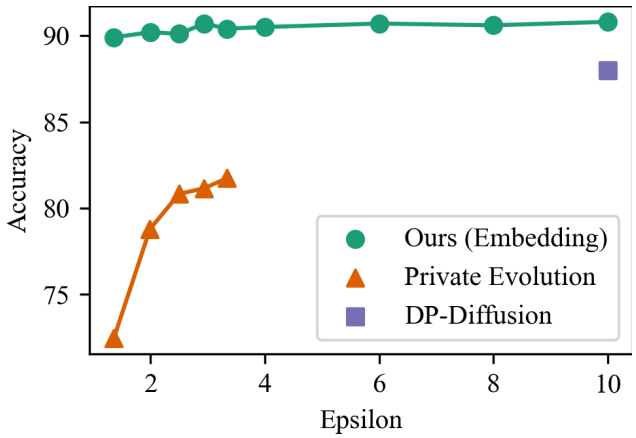

Fig. 2: Downstream classification accuracy on 50,000 generated CIFAR-10 images at various levels of ε and $\delta = 10^{-5}$. We report the baseline accuracies at every available level of ε, exactly as stated in their respective papers.

Table 2: Downstream classification accuracies when trained on 50,000 synthetic embeddings or images and tested on CIFAR-10. $\delta = 10^{-5}$ for all experiments.

ε	DP-SGD (fine-tuning)	Ours (embeddings)	Ours (images)
1	94.8	96.6 ± 0.074	89.7 ± 0.143
2	95.4	96.8 ± 0.087	90.2 ± 0.129
4	96.1	96.9 ± 0.064	90.5 ± 0.044
8	96.6	97.0 ± 0.01	90.6 ± 0.054

accuracy does not significantly decrease as ε varies, the variance of the generated images noticeably increases as ε decreases. Consider for example, the 9-th row of Figures 3 and 6, which displays synthetic images of boats at $\varepsilon = 8, 1$, respectively. In Figure 3, each of the 10 images is recognizable as a boat. However, in Figure 6, only 3 of the 10 images resemble some form of a boat while the others in the row are abstract shapes. Similar occurrences can be observed for the other classes

4 Limitations & Future Works

DP clustering. An important subroutine in our generation pipeline is DP clustering. Improved implementations of DP clustering can also improve our algorithm, further motivating research on DP clustering.

Decoding. We were unable to find other encoder-decoder pairs that generalize beyond their training data, which necessitated the use of CLIP embeddings if we wish to generate images. Progress on general-purpose encoder-decoder models or exploration of domain-specific encoder-decoder pairs will broaden the applicability of our method.

Filtering. The generated images were not carefully filtered, and we used two simple filtering heuristics that are agnostic to the sensitive data. Using more sophisticated methods [EM18; KWW+21] may yield better performance. Furthermore, we can use existing image enhancement methods [QYS+21] to improve the quality of generated images.

5 Conclusion

Our work introduces a novel principled framework for private training and data generation by clustering embeddings, demonstrating significant improvements in privacy-utility tradeoffs compared to existing approaches in the DP synthetic data literature. Moreover, by leveraging DP synthetic embeddings, we achieve state-of-the-art classification accuracy on CIFAR-10 and CAMELYON17, highlighting the potential of our method in real-world applications. Our method offers a practical solution for privately training classifiers without exposing real data, which is particularly valuable in domains where data sharing is restricted.

Acknowledgments

This work was partially completed while Felix Zhou was a student researcher at Google Research. Felix Zhou acknowledges the support of the Natural Sciences and Engineering Research Council of Canada (NSERC). Samson Zhou is supported in part by NSF CCF-2335411.

References

[AAB+25] Kareem Amin, Salman Avestimehr, Sara Babakniya, Alex Bie, Weiwei Kong, Natalia Ponomareva, and Umar Syed. "Clustering and Median Aggregation Improve Differentially Private Inference". In: *arXiv preprint arXiv:2506.04566* (2025) (cit. on p. 2).

[AAK21] Ishaq Aden-Ali, Hassan Ashtiani, and Gautam Kamath. "On the Sample Complexity of Privately Learning Unbounded High-Dimensional Gaussians". In: *Algorithmic Learning Theory, 16-19 March 2021, Virtual Conference, Worldwide*. 2021, pp. 185–216 (cit. on p. 31).

[AAL23] Jamil Arbas, Hassan Ashtiani, and Christopher Liaw. "Polynomial Time and Private Learning of Unbounded Gaussian Mixture Models". In: *International Conference on Machine Learning, ICML*. 2023, pp. 1018–1040 (cit. on p. 6).

[ACG+16] Martín Abadi, Andy Chu, Ian J. Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. "Deep Learning with Differential Privacy". In: *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*. 2016, pp. 308–318 (cit. on pp. 2, 26).

[ACH+24] Kyriakos Axiotis, Vincent Cohen-Addad, Monika Henzinger, Sammy Jerome, Vahab Mirrokni, David Saulpic, David P. Woodruff, and Michael Wunder. "Data-Efficient Learning via Clustering-Based Sensitivity Sampling: Foundation Models and Beyond". In: *Forty-first International Conference on Machine Learning, ICML*. 2024 (cit. on pp. 4, 29).

[AK05] Sanjeev Arora and Ravi Kannan. “Learning mixtures of separated nonspherical Gaussians”. In: (2005) (cit. on p. 6).

[Ale09] Krizhevsky Alex. “Learning multiple layers of features from tiny images”. In: *https://www. cs. toronto. edu/kriz/learning-features-2009-TR. pdf* (2009) (cit. on pp. 9, 10).

[ASE17] Antreas Antoniou, Amos J. Storkey, and Harrison Edwards. “Data Augmentation Generative Adversarial Networks”. In: *CoRR* abs/1711.04340 (2017) (cit. on p. 26).

[AV18] Pranjal Awasthi and Aravindan Vijayaraghavan. “Clustering Semi-Random Mixtures of Gaussians”. In: *Proceedings of the 35th International Conference on Machine Learning, ICML*. 2018, pp. 294–303 (cit. on p. 34).

[AZK+18] Nazmiye Ceren Abay, Yan Zhou, Murat Kantarcioglu, Bhavani Thuraisingham, and Latanya Sweeney. “Privacy Preserving Synthetic Data Release Using Deep Learning”. In: *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD, Proceedings, Part I*. 2018, pp. 510–526 (cit. on pp. 2, 5).

[BDK+20] Sourav Biswas, Yihe Dong, Gautam Kamath, and Jonathan R. Ullman. “CoinPress: Practical Private Mean and Covariance Estimation”. In: *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems, NeurIPS*. 2020 (cit. on p. 31).

[BGM+19] Péter Bándi, Oscar Geessink, Quirine Manson, Marcory Van Dijk, Maschenka Balkenhol, Meyke Hermsen, Babak Ehteshami Bejnordi, Byungjae Lee, Kyunghyun Paeng, Aoxiao Zhong, Quanzheng Li, Farhad Ghazvinian Zanjani, Svitlana Zinger, Keisuke Fukuta, Daisuke Komura, Vlado Ovtcharov, Shenghua Cheng, Shaoqun Zeng, Jeppe Thagaard, Anders B. Dahl, Huangjing Lin, Hao Chen, Ludwig Jacobsson, Martin Hedlund, Melih Çetin, Eren Halici, Hunter Jackson, Richard Chen, Fabian Both, Jörg Franke, Heidi Küsters-Vandevelde, Willem Vreuls, Peter Bult, Bram van Ginneken, Jeroen van der Laak, and Geert Litjens. “From Detection of Individual Metastases to Classification of Lymph Node Status at the Patient Level: The CAMELYON17 Challenge”. In: *IEEE Trans. Medical Imaging* 38.2 (2019), pp. 550–560 (cit. on pp. 9, 10).

[CD18] Rachel Cummings and Deven Desai. “The role of differential privacy in gdpr compliance”. In: *FAT’18: Proceedings of the Conference on Fairness, Accountability, and Transparency*. Vol. 20. 2018 (cit. on p. 2).

[CEL+22] Vincent Cohen-Addad, Alessandro Epasto, Silvio Lattanzi, Vahab Mirrokni, Andres Muñoz Medina, David Saulpic, Chris Schwiegelshohn, and Sergei Vassilvitskii. “Scalable Differentially Private Clustering via Hierarchically Separated Trees”. In: *KDD ’22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2022, pp. 221–230 (cit. on pp. 6, 33).

[CEM+22] Vincent Cohen-Addad, Alessandro Epasto, Vahab Mirrokni, Shyam Narayanan, and Peilin Zhong. “Near-Optimal Private and Scalable k-Clustering”. In: *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems, NeurIPS*. 2022 (cit. on pp. 6, 33).

[CK21] Alisa Chang and Pritish Kamath. *Differentially private clustering in google's differential privacy library*. https://github.com/google/differential-privacy/tree/main/learning/clustering. Accompanying article available at https://research.google/blog/practical-differentially-private-clustering/. 2021 (cit. on p. 8).

[CKF24] Dingfan Chen, Raouf Kerkouche, and Mario Fritz. "A Unified View of Differentially Private Deep Generative Modeling". In: *Trans. Mach. Learn. Res.* 2024 (2024) (cit. on p. 5).

[Das04] Sanjoy Dasgupta. "Analysis of a greedy active learning strategy". In: *Advances in Neural Information Processing Systems 17 [Neural Information Processing Systems, NIPS*. 2004, pp. 337–344 (cit. on p. 27).

[Das99] Sanjoy Dasgupta. "Learning Mixtures of Gaussians". In: *40th Annual Symposium on Foundations of Computer Science, FOCS '99, 17-18 October, 1999, New York, NY, USA*. IEEE Computer Society, 1999, pp. 634–644. DOI: 10.1109/SFFCS.1999.814639. URL: https://doi.org/10.1109/SFFCS.1999.814639 (cit. on pp. 4, 6).

[DBH+22] Soham De, Leonard Berrada, Jamie Hayes, Samuel L Smith, and Borja Balle. "Unlocking high-accuracy differentially private image classification through scale". In: *arXiv preprint arXiv:2204.13650* (2022) (cit. on pp. 2, 5, 10, 11).

[DBK+21] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *9th International Conference on Learning Representations, ICLR*. 2021 (cit. on p. 11).

[DCL+19a] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*. 2019, pp. 4171–4186 (cit. on pp. 4, 7).

[DCL+19b] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*. 2019, pp. 4171–4186 (cit. on p. 4).

[DDS+09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848 (cit. on pp. 9–11).

[DKW56] Aryeh Dvoretzky, Jack Kiefer, and Jacob Wolfowitz. "Asymptotic minimax character of the sample distribution function and of the classical multinomial estimator". In: *The Annals of Mathematical Statistics* (1956), pp. 642–669 (cit. on p. 30).

[DLR77] Arthur P Dempster, Nan M Laird, and Donald B Rubin. "Maximum likelihood from incomplete data via the EM algorithm". In: *Journal of the royal statistical society: series B (methodological)* 39.1 (1977), pp. 1–22 (cit. on p. 6).

[DMN+06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith. "Calibrating Noise to Sensitivity in Private Data Analysis". In: *Theory of Cryptography, Third Theory of Cryptography Conference, TCC, Proceedings*. 2006, pp. 265–284 (cit. on pp. 2, 29).

[DR14] Cynthia Dwork and Aaron Roth. "The Algorithmic Foundations of Differential Privacy". In: *Found. Trends Theor. Comput. Sci.* 9.3-4 (2014), pp. 211–407 (cit. on pp. 29, 30).

[DS00] Sanjoy Dasgupta and Leonard J. Schulman. "A Two-Round Variant of EM for Gaussian Mixtures". In: *UAI '00: Proceedings of the 16th Conference in Uncertainty in Artificial Intelligence, Stanford University, Stanford, California, USA, June 30 - July 3, 2000*. Ed. by Craig Boutilier and Moisés Goldszmidt. Morgan Kaufmann, 2000, pp. 152–159. URL: https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1%5C&smnu=2%5C&article%5C_id=18%5C&proceeding%5C_id=16 (cit. on p. 6).

[EM18] Hossein Talebi Esfandarani and Peyman Milanfar. "NIMA: Neural Image Assessment". In: *IEEE Trans. Image Process.* 27.8 (2018), pp. 3998–4011 (cit. on p. 13).

[FDK+18] Maayan Frid-Adar, Idit Diamant, Eyal Klang, Michal Amitai, Jacob Goldberger, and Hayit Greenspan. "GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification". In: *Neurocomputing* 321 (2018), pp. 321–331 (cit. on p. 26).

[FKM+21] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. "Sharpness-aware Minimization for Efficiently Improving Generalization". In: *9th International Conference on Learning Representations, ICLR*. 2021 (cit. on p. 11).

[GBG+23] Sahra Ghalebikesabi, Leonard Berrada, Sven Gowal, Ira Ktena, Robert Stanforth, Jamie Hayes, Soham De, Samuel L. Smith, Olivia Wiles, and Borja Balle. "Differentially Private Diffusion Models Generate Useful Synthetic Images". In: *CoRR* abs/2302.13861 (2023) (cit. on pp. 2, 5, 10, 11).

[GL16] Aditya Grover and Jure Leskovec. "node2vec: Scalable Feature Learning for Networks". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2016, pp. 855–864 (cit. on p. 4).

[GPM+14] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems*. 2014, pp. 2672–2680 (cit. on p. 26).

[GRG06] Hayit Greenspan, Amit Ruf, and Jacob Goldberger. "Constrained Gaussian mixture model framework for automatic segmentation of MR brain images". In: *IEEE transactions on medical imaging* 25.9 (2006), pp. 1233–1245 (cit. on p. 6).

[HHW+14] Peihao Huang, Yan Huang, Wei Wang, and Liang Wang. "Deep Embedding Network for Clustering". In: *22nd International Conference on Pattern Recognition, ICPR 2014, Stockholm, Sweden, August 24-28, 2014*. IEEE Computer Society, 2014, pp. 1532–1537 (cit. on p. 3).

[HJS+23] Frederik Harder, Milad Jalali, Danica J. Sutherland, and Mijung Park. "Pre-trained Perceptual Features Improve Differentially Private Image Generation". In: *Trans. Mach. Learn. Res.* 2023 (2023) (cit. on pp. 2, 5, 11).

[HK13] Daniel Hsu and Sham M Kakade. "Learning mixtures of spherical gaussians: moment methods and spectral decompositions". In: *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*. 2013, pp. 11–20 (cit. on p. 6).

[HKM+23] Samuel B. Hopkins, Gautam Kamath, Mahbod Majid, and Shyam Narayanan. "Robustness Implies Privacy in Statistical Estimation". In: *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC*. 2023, pp. 497–506 (cit. on p. 31).

[HKM22] Samuel B. Hopkins, Gautam Kamath, and Mahbod Majid. "Efficient mean estimation with pure differential privacy via a sum-of-squares exponential mechanism". In: *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing*. ACM, 2022, pp. 1406–1417 (cit. on p. 31).

[HL18] Zhiyi Huang and Jinyan Liu. "Optimal Differentially Private Algorithms for k-Means Clustering". In: *Proceedings of the 37th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*. 2018, pp. 395–408 (cit. on p. 6).

[HLY+23] Jiyan He, Xuechen Li, Da Yu, Huishuai Zhang, Janardhan Kulkarni, Yin Tat Lee, Arturs Backurs, Nenghai Yu, and Jiang Bian. "Exploring the Limits of Differentially Private Deep Learning with Group-wise Clipping". In: *The Eleventh International Conference on Learning Representations, ICLR*. 2023 (cit. on p. 2).

[HSZ+24] Charlie Hou, Akshat Shrivastava, Hongyuan Zhan, Rylan Conway, Trang Le, Adithya Sagar, Giulia Fanti, and Daniel Lazar. "PrE-Text: Training Language Models on Private Federated Data in the Age of LLMs". In: *Forty-first International Conference on Machine Learning, ICML*. 2024 (cit. on pp. 2, 5, 7).

[HW71] David Lee Hanson and Farroll Tim Wright. "A bound on tail probabilities for quadratic forms in independent random variables". In: *The Annals of Mathematical Statistics* 42.3 (1971), pp. 1079–1083 (cit. on p. 30).

[HWL+24] Yuzheng Hu, Fan Wu, Qinbin Li, Yunhui Long, Gonzalo Munilla Garrido, Chang Ge, Bolin Ding, David A. Forsyth, Bo Li, and Dawn Song. "SoK: Privacy-Preserving Data Synthesis". In: *IEEE Symposium on Security and Privacy, SP*. 2024, pp. 4696–4713 (cit. on p. 5).

[HZR+16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*. IEEE Computer Society, 2016, pp. 770–778 (cit. on pp. 4, 9, 11, 28).

[JZT+17] Zhuxi Jiang, Yin Zheng, Huachun Tan, Bangsheng Tang, and Hanning Zhou. "Variational Deep Embedding: An Unsupervised and Generative Approach to Clustering". In: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI*. ijcai.org, 2017, pp. 1965–1972 (cit. on p. 3).

[KF18] Angelos Katharopoulos and François Fleuret. "Not All Samples Are Created Equal: Deep Learning with Importance Sampling". In: *Proceedings of the 35th International Conference on Machine Learning, ICML*. 2018, pp. 2530–2539 (cit. on p. 27).

[KMS22] Gautam Kamath, Argyris Mouzakis, and Vikrant Singhal. "New Lower Bounds for Private Estimation and a Generalized Fingerprinting Lemma". In: *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems, NeurIPS*. 2022 (cit. on p. 31).

[KMV10] Adam Tauman Kalai, Ankur Moitra, and Gregory Valiant. "Efficiently learning mixtures of two Gaussians". In: *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*. Ed. by Leonard J. Schulman. ACM, 2010, pp. 553–562. DOI: 10.1145/1806689.1806765. URL: https://doi.org/10.1145/1806689.1806765 (cit. on p. 6).

[KPS+23] Alexey Kurakin, Natalia Ponomareva, Umar Syed, Liam MacDermed, and Andreas Terzis. "Harnessing large-language models to generate private synthetic text". In: *arXiv preprint arXiv:2306.01684* (2023) (cit. on pp. 2, 5).

[KSM+21] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton Earnshaw, Imran S. Haque, Sara M. Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. "WILDS: A Benchmark of in-the-Wild Distribution Shifts". In: *Proceedings of the 38th International Conference on Machine Learning, ICML*. 2021, pp. 5637–5664 (cit. on p. 27).

[KSS+19] Gautam Kamath, Or Sheffet, Vikrant Singhal, and Jonathan R. Ullman. "Differentially Private Algorithms for Learning Mixtures of Separated Gaussians". In: *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems, NeurIPS*. 2019, pp. 168–180 (cit. on p. 6).

[KV18] Vishesh Karwa and Salil P. Vadhan. "Finite Sample Differentially Private Confidence Intervals". In: *9th Innovations in Theoretical Computer Science Conference, ITCS*. 2018, 44:1–44:9 (cit. on p. 31).

[KWW+21] Junjie Ke, Qifei Wang, Yilin Wang, Peyman Milanfar, and Feng Yang. "MUSIQ: Multi-scale Image Quality Transformer". In: *2021 IEEE/CVF International Conference on Computer Vision, ICCV*. 2021, pp. 5128–5137 (cit. on p. 13).

[LGK+24] Zinan Lin, Sivakanth Gopi, Janardhan Kulkarni, Harsha Nori, and Sergey Yekhanin. "Differentially Private Synthetic Data via Foundation Model APIs 1: Images". In: *The Twelfth International Conference on Learning Representations, ICLR*. 2024 (cit. on pp. 2, 5, 7, 8, 10, 11, 27).

[LOC+19] Sanghun Lee, Sangjun Oh, Kyuhyoung Choi, and S Kim. "Automatic classification on patient-level breast cancer metastases". In: *Submission Results Camelyon17 Challenge* (2019) (cit. on p. 11).

[LTH+17] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew P. Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*. IEEE Computer Society, 2017, pp. 105–114 (cit. on p. 26).

[Lux07] Ulrike von Luxburg. "A tutorial on spectral clustering". In: *Stat. Comput.* 17.4 (2007), pp. 395–416 (cit. on p. 3).

[LWG+19] Zefang Lv, Lirong Wang, Zhitao Guan, Jun Wu, Xiaojiang Du, Hongtao Zhao, and Mohsen Guizani. “An Optimizing and Differentially Private Clustering Algorithm for Mixed Data in SDN-Based Smart Grid”. In: *IEEE Access* 7 (2019), pp. 45773–45782 (cit. on p. 6).

[Mas90] Pascal Massart. “The tight constant in the Dvoretzky-Kiefer-Wolfowitz inequality”. In: *The annals of Probability* (1990), pp. 1269–1283 (cit. on p. 30).

[MC16] Maintainers and Contributors. *TorchVision: PyTorch’s Computer Vision library*. https://github.com/pytorch/vision. 2016 (cit. on pp. 9, 28).

[MCC+13] Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. “Efficient Estimation of Word Representations in Vector Space”. In: *1st International Conference on Learning Representations, ICLR*. 2013 (cit. on p. 4).

[McS09] Frank McSherry. “Privacy integrated queries: an extensible platform for privacy-preserving data analysis”. In: *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD*. ACM, 2009, pp. 19–30 (cit. on p. 30).

[MGN+23] Sadhika Malladi, Tianyu Gao, Eshaan Nichani, Alex Damian, Jason D. Lee, Danqi Chen, and Sanjeev Arora. “Fine-Tuning Language Models with Just Forward Passes”. In: *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems, NeurIPS*. 2023 (cit. on p. 2).

[MJW+22] Justus Mattern, Zhijing Jin, Benjamin Weggenmann, Bernhard Schölkopf, and Mrinmaya Sachan. “Differentially Private Language Models for Secure Data Sharing”. In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP*. Association for Computational Linguistics, 2022, pp. 4860–4873 (cit. on pp. 2, 5).

[Moi18] Ankur Moitra. *Algorithmic aspects of machine learning*. Cambridge University Press, 2018 (cit. on p. 6).

[MSB13] Anish Mittal, Rajiv Soundararajan, and Alan C. Bovik. “Making a "Completely Blind" Image Quality Analyzer”. In: *IEEE Signal Process. Lett.* 20.3 (2013), pp. 209–212 (cit. on pp. 9, 28).

[MTK+17] Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. “Pruning Convolutional Neural Networks for Resource Efficient Inference”. In: *5th International Conference on Learning Representations, ICLR*. 2017 (cit. on p. 27).

[MU05] Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, 2005 (cit. on p. 30).

[MV10] Ankur Moitra and Gregory Valiant. “Settling the Polynomial Learnability of Mixtures of Gaussians”. In: *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*. IEEE Computer Society, 2010, pp. 93–102. DOI: 10.1109/FOCS.2010.15. URL: https://doi.org/10.1109/FOCS.2010.15 (cit. on p. 6).

[Nar24] Shyam Narayanan. “Better and simpler lower bounds for differentially private statistical estimation”. In: *IEEE Transactions on Information Theory* (2024) (cit. on p. 31).

[Pea94] Karl Pearson. "Contributions to the mathematical theory of evolution". In: *Philosophical Transactions of the Royal Society of London. A* 185 (1894), pp. 71–110 (cit. on p. 6).

[PFC+17] Mijung Park, James R. Foulds, Kamalika Choudhary, and Max Welling. "DP-EM: Differentially Private Expectation Maximization". In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS*. 2017, pp. 896–904 (cit. on p. 6).

[PH24] Victor S. Portella and Nick Harvey. "Lower Bounds for Private Estimation of Gaussian Covariance Matrices under All Reasonable Parameter Regimes". In: *CoRR* abs/2404.17714 (2024) (cit. on p. 31).

[PSM14] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. "Glove: Global Vectors for Word Representation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP*. 2014, pp. 1532–1543 (cit. on p. 4).

[PY10] Sinno Jialin Pan and Qiang Yang. "A Survey on Transfer Learning". In: *IEEE Trans. Knowl. Data Eng.* 22.10 (2010), pp. 1345–1359 (cit. on p. 27).

[QYS+21] Yunliang Qi, Zhen Yang, Wenhao Sun, Meng Lou, Jing Lian, Wenwei Zhao, Xiangyu Deng, and Yide Ma. "A comprehensive overview of image enhancement techniques". In: *Archives of Computational Methods in Engineering* (2021), pp. 1–25 (cit. on p. 13).

[RAY+16] Scott E. Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. "Generative Adversarial Text to Image Synthesis". In: *Proceedings of the 33nd International Conference on Machine Learning, ICML*. 2016, pp. 1060–1069 (cit. on p. 26).

[RBL+22] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. "High-Resolution Image Synthesis With Latent Diffusion Models". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 10684–10695 (cit. on pp. 9, 10).

[RDS+19] Benedek Rozemberczki, Ryan Davies, Rik Sarkar, and Charles Sutton. "GEMSEC: graph embedding with self clustering". In: *ASONAM '19: International Conference on Advances in Social Networks Analysis and Mining, Vancouver, British Columbia, Canada, 27-30 August, 2019*. ACM, 2019, pp. 65–72 (cit. on p. 3).

[Rey+09] Douglas A Reynolds et al. "Gaussian mixture models." In: *Encyclopedia of biometrics* 741.659-663 (2009), p. 3 (cit. on pp. 6, 29).

[RKH+21] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. "Learning Transferable Visual Models From Natural Language Supervision". In: *Proceedings of the 38th International Conference on Machine Learning, ICML*. 2021, pp. 8748–8763 (cit. on pp. 4, 8, 10).

[RLP+20] Lucas Rosenblatt, Xiaoyan Liu, Samira Pouyanfar, Eduardo de Leon, Anuj Desai, and Joshua Allen. "Differentially private synthetic data: Applied evaluations and enhancements". In: *arXiv preprint arXiv:2011.05537* (2020) (cit. on pp. 2, 5).

[RMC16] Alec Radford, Luke Metz, and Soumith Chintala. "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks". In: *4th International Conference on Learning Representations, ICLR*. 2016 (cit. on p. 26).

[RSB+19] Nils Reimers, Benjamin Schiller, Tilman Beck, Johannes Daxenberger, Christian Stab, and Iryna Gurevych. "Classification and Clustering of Arguments with Contextualized Word Embeddings". In: *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL*. Association for Computational Linguistics, 2019, pp. 567–578 (cit. on p. 3).

[RSR+20] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. "Exploring the limits of transfer learning with a unified text-to-text transformer". In: *Journal of machine learning research* 21.140 (2020), pp. 1–67 (cit. on pp. 4, 7).

[RXC+22] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B. Gupta, Xiaojiang Chen, and Xin Wang. "A Survey of Deep Active Learning". In: *ACM Comput. Surv.* 54.9 (2022), 180:1–180:40 (cit. on p. 27).

[SBV+22] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. "LAION-5B: An open large-scale dataset for training next generation image-text models". In: *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems, NeurIPS*. 2022 (cit. on p. 10).

[SCL+16] Dong Su, Jianneng Cao, Ninghui Li, Elisa Bertino, and Hongxia Jin. "Differentially Private K-Means Clustering". In: *Proceedings of the Sixth ACM on Conference on Data and Application Security and Privacy, CODASPY*. 2016, pp. 26–37 (cit. on p. 6).

[SCL+17] Dong Su, Jianneng Cao, Ninghui Li, Elisa Bertino, Min Lyu, and Hongxia Jin. "Differentially Private K-Means Clustering and a Hybrid Approach to Private Optimization". In: *ACM Trans. Priv. Secur.* 20.4 (2017), 16:1–16:33 (cit. on p. 6).

[Set09] Burr Settles. *Active learning literature survey*. 2009 (cit. on p. 27).

[SGG16] Abhinav Shrivastava, Abhinav Gupta, and Ross B. Girshick. "Training Region-Based Object Detectors with Online Hard Example Mining". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*. 2016, pp. 761–769 (cit. on p. 27).

[She05] H Sheikh. "LIVE image quality assessment database release 2". In: *http://live.ece.utexas.edu/research/quality* (2005) (cit. on pp. 9, 28).

[SK19] Connor Shorten and Taghi M. Khoshgoftaar. "A survey on Image Data Augmentation for Deep Learning". In: *J. Big Data* 6 (2019), p. 60 (cit. on p. 27).

[SM21] Liwei Song and Prateek Mittal. "Systematic evaluation of privacy risks of machine learning models". In: *30th USENIX Security Symposium (USENIX Security 21)*. 2021, pp. 2615–2632 (cit. on p. 1).

[Spi25] DA Spielman. *Spectral and Algebraic Graph Theory, Book draft*. 2025 (cit. on p. 3).

[SS18] Ozan Sener and Silvio Savarese. "Active Learning for Convolutional Neural Networks: A Core-Set Approach". In: *6th International Conference on Learning Representations, ICLR, Conference Track Proceedings*. 2018 (cit. on p. 4).

[SYP+19] Veit Sandfort, Ke Yan, Perry J Pickhardt, and Ronald M Summers. "Data augmentation using generative adversarial networks (CycleGAN) to improve generalizability in CT segmentation tasks". In: *Scientific reports* 9.1 (2019), p. 16884 (cit. on p. 26).

[SZ15] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *3rd International Conference on Learning Representations, ICLR*. 2015 (cit. on p. 4).

[TFR22] Amirsina Torfi, Edward A. Fox, and Chandan K. Reddy. "Differentially private synthetic medical data generation using convolutional GANs". In: *Inf. Sci.* 586 (2022), pp. 485–500 (cit. on pp. 2, 5).

[TKP19] Reihaneh Torkzadehmahani, Peter Kairouz, and Benedict Paten. "DP-CGAN: Differentially Private Synthetic Data and Label Generation". In: *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops*. Computer Vision Foundation / IEEE, 2019, pp. 98–104 (cit. on pp. 2, 5, 10).

[TXX+25] Bowen Tan, Zheng Xu, Eric Xing, Zhiting Hu, and Shanshan Wu. "Synthesizing Privacy-Preserving Text Data via Finetuning without Finetuning Billion-Scale LLMs". In: *CoRR* abs/2503.12347 (2025) (cit. on p. 2).

[Ver10] Roman Vershynin. "Introduction to the non-asymptotic analysis of random matrices". In: *arXiv preprint arXiv:1011.3027* (2010) (cit. on p. 30).

[VPB+15] Narasimhan Venkatanath, D Praneeth, Maruthi Chandrasekhar Bh, Sumohana S Channappayya, and Swarup S Medasani. "Blind image quality evaluation using perception based features". In: *Twenty First National Conference on Communications, NCC*. IEEE, 2015, pp. 1–6 (cit. on pp. 9, 28).

[VV17] Paul Voigt and Axel Von dem Bussche. "The eu general data protection regulation (gdpr)". In: *A practical guide, 1st ed., Cham: Springer International Publishing* 10.3152676 (2017), pp. 10–5555 (cit. on p. 1).

[VW04] Santosh S. Vempala and Grant Wang. "A spectral algorithm for learning mixture models". In: *J. Comput. Syst. Sci.* 68.4 (2004), pp. 841–860. DOI: 10.1016/J.JCSS.2003.11.008. URL: https://doi.org/10.1016/j.jcss.2003.11.008 (cit. on p. 6).

[Wai19] Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*. Vol. 48. Cambridge university press, 2019 (cit. on p. 4).

[XGF16] Junyuan Xie, Ross B. Girshick, and Ali Farhadi. "Unsupervised Deep Embedding for Clustering Analysis". In: *Proceedings of the 33nd International Conference on Machine Learning, ICML*. Vol. 48. JMLR Workshop and Conference Proceedings. 2016, pp. 478–487 (cit. on p. 3).

[XLB+24] Chulin Xie, Zinan Lin, Arturs Backurs, Sivakanth Gopi, Da Yu, Huseyin A. Inan, Harsha Nori, Haotian Jiang, Huishuai Zhang, Yin Tat Lee, Bo Li, and Sergey Yekhanin. "Differentially Private Synthetic Data via Foundation Model APIs 2: Text". In: *Forty-first International Conference on Machine Learning, ICML*. 2024 (cit. on pp. 2, 5, 7, 8).

[YIL+23] Xiang Yue, Huseyin A. Inan, Xuechen Li, Girish Kumar, Julia McAnallen, Hoda Shajari, Huan Sun, David Levitan, and Robert Sim. "Synthetic Text Generation with Differential Privacy: A Simple and Practical Recipe". In: *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL*. 2023, pp. 1321–1342 (cit. on pp. 2, 5).

[YLL+14] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z. Li. "Learning Face Representation from Scratch". In: *CoRR* abs/1411.7923 (2014) (cit. on p. 26).

[YNB+22] Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A. Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, Sergey Yekhanin, and Huishuai Zhang. "Differentially Private Fine-tuning of Language Models". In: *The Tenth International Conference on Learning Representations, ICLR*. 2022 (cit. on pp. 2, 26).

[YSS+21] Ashkan Yousefpour, Igor Shilov, Alexandre Sablayrolles, Davide Testuggine, Karthik Prasad, Mani Malek, John Nguyen, Sayan Ghosh, Akash Bharadwaj, Jessica Zhao, et al. "Opacus: User-friendly differential privacy library in PyTorch". In: *arXiv preprint arXiv:2109.12298* (2021) (cit. on p. 3).

A CIFAR-10 Synthetic Images

In this section, we show examples of synthetic CIFAR-10 images generated at various levels of ε. We randomly chose 10 images per class from each of the synthetic and original training sets and display them side-by-side. As expected, there is a noticeable decrease in the fidelity of the synthetic images as ε decreases, due to the increase in noise injected into the system.

Fig. 3: CIFAR-10 synthetic images at $\varepsilon = 8, \delta = 10^{-5}$. Each row corresponds to a different class. The left-most columns are synthetic images while the right-most columns are original images.

Fig. 4: CIFAR-10 synthetic images at $\varepsilon = 4, \delta = 10^{-5}$. Each row corresponds to a different class. The left-most columns are synthetic images while the right-most columns are original images.

Fig. 5: CIFAR-10 synthetic images at $\varepsilon = 2, \delta = 10^{-5}$. Each row corresponds to a different class. The left-most columns are synthetic images while the right-most columns are original images.

Fig. 6: CIFAR-10 synthetic images at $(\varepsilon = 1, \delta = 10^{-5})$. Each row corresponds to a different class. The left-most columns are synthetic images while the right-most columns are original images.

B Related Work

The areas most related to our work are that of data selection, (non-private) synthetic data generation, and private fine-tuning.

Synthetic data generation. The seminal paper of Goodfellow, Pouget-Abadie, Mirza, et al. [GPM+14] introduced Generative Adversarial Networks (GANs), which intuitively train the generator not to minimize the loss function for labels of individual images, but instead to fool a "discriminator" neural network that can tell how "realistic" the input seems. GANs have been widely used for generating synthetic data [RMC16] across a wide range of applications, e.g., to privately create synthetic medical images to enhance CNN performance [FDK+18; SYP+19] in healthcare, as well as for applications in data augmentation [ASE17], facial recognition [YLL+14], image super-resolution [LTH+17], and text-to-image synthesis [RAY+16].

Private fine-tuning. Another relevant area is that of private fine-tuning, which has been used for tasks such as privately training large language models [YNB+22]. Private fine-tuning is the process of adapting a pre-trained machine learning model to a specific task using a sensitive dataset, while ensuring that individual data points in the dataset remain private. The main intuition behind private fine-tuning is to modify the model's training process with private techniques to prevent the model from compromising sensitive information, e.g., using methods such as DP-SGD [ACG+16], which adds noise to the gradients during training and clips them to control the influence of any single data point. Consequently, the resulting model retains useful task-specific knowledge while preserving differential privacy.

Recently, Lin, Gopi, Kulkarni, et al. [LGK+24] observed that API-based solutions are becoming increasingly popular, in part due to the accessibility of these systems to users without ML-specific expertise. Thus, they view API providers as untrusted entities and proposed the Private Evolution algorithm for generating private synthetic data using black-box APIs of foundation models. Despite not having access to model weights and gradients, the key idea of Private Evolution is to iteratively utilize private samples to determine the most similar samples generated from the black-box model and ask the black-box models to generate more of those similar samples.

Data selection. Data selection aims to select the most "important" data points to label from a pool of unlabeled examples, thereby maximizing accuracy with a smaller labeled dataset, reducing the cost and effort of labeling while still achieving high model performance. While a universally optimal data selection strategy is not achievable [Das04], several heuristics [Set09; RXC+22] have proven to be effective in practice. Nevertheless, these traditional data selection approaches all focus on selecting the most important data points to improve model performance. Indeed, data selection inherently and necessarily reveals crucial structural information about these data points that is subsequently used for generalization.

Data selection for machine learning is a well-studied approach for improving model efficiency, robustness, and generalization, particularly in the context of deep learning and neural networks. Perhaps the most relevant technique to our line of study is importance sampling, where training emphasizes samples that contribute most to the learning objective, enhancing gradient efficiency and thus convergence rates [KF18]. Similarly, data pruning methods seek to remove redundant or less informative samples in the training data, while retaining critical patterns, thereby lowering computational costs without sacrificing accuracy [MTK+17]. For CNNs, where spatial structure in data is crucial, hard example mining [SGG16] is often employed to focus training on misclassified or challenging samples, which helps the model learn more expressive features. Other data augmentation and selection methods, such as diversity-based selection, aim to include a wide range of spatial and contextual patterns, to reduce overfitting and improve robustness [SK19]. On the other hand, task-specific methods, such as domain-aware selection in transfer learning, prioritize source domain samples similar to the target domain, enabling better feature transfer [PY10].

C Further Experimental Details

C.1 Datasets

CIFAR10 consists of 60,000 32×32 natural images in 10 equal-sized classes. The standard training split consists of 50,000 images.

CIFAR100 is similar and consists of 60,000 32×32 natural images in 100 equal-sized classes. The training split consists again of 50,000 images. This is a challenging dataset for DP synthetic data as each class has only 500 training images and different classes can have very similar appearances.

Finally, CAMELYON17 is a medical dataset for classification of breast cancer metastases. It consists of 96×96 image patches of lymph node tissue from five different hospitals. The label signifies whether at least one pixel in the center 32×32 pixels has been identified as a tumor cell. CAMELYON17 is part of the WILDS [KSM+21] leaderboard as a domain generalization task: The training split contains 302,436 images from three different hospitals whereas the test split contain 85,054 images from a fourth and fifth hospital.

C.2 Generation Details

We perform a gridsearch over the number of k of GMM components, and the intra-cluster clipping radius for covariance estimation. We choose $k \in \{1, 2, 4, 8, 16\}$ and the clipping radius between $\{2.0, 4.0, 6.0, 8.0, 10.0\}$.

We tried various covariance models of GMMs and found that diagonal covariances yielded the best performance, as spherical covariance models do not capture enough of the intra-cluster data and the noise needed to privately estimate a full covariance matrix overwhelms the signal from the data.

When pruning embeddings, we discard all generated embeddings that do not have a noisy vote of at least 6.0. If the number of images per class is m, we generate $6m$ synthetic embeddings and find that approximately $2m$ embeddings survive.

When decoding images, we process embeddings in batches of 16 per GPU and generate 2 images per embedding. We discard all images whose NIQE [MSB13] or PIQE [She05; VPB+15] score falls below 20.0. We found that approximately m images per class survive this process.

C.3 Embedding Classification Details

We train a simple two-layer neural network with a 128-dimensional hidden layer, batch normalization, and dropout probability 0.5.

We train on a single GPU with a batch size of 512 for 50 epochs. We use SGD with an initial learning rate of 10^{-3} and cosine annealing learning rate scheduler. For further regularization, we set label smoothing to 0.2 and weight decay to 10^{-4}.

Before training, if there are more than n synthetic embeddings where n is the size of the input sensitive training set, then we select a random subset of size n. This is to maintain fair comparison against private training baselines that do not use synthetic data and therefore only have access to n training points.

C.4 Image Classification Details

We fine-tune the torchvision implementation of ResNet50 [HZR+16] which was pre-trained on ImageNet [HZR+16] with pre-trained weights publicly available from torchvision [MC16].

We train on a cluster of eight H100 GPUs (80GB memory each) with a batch size of 256 per GPU for 30 epochs. As mentioned, we follow the torchvision training recipe.[7] The only change is the initial learning rate of 10^{-2} and a pre-processing step where we encode and decode the test set (without using it in the training process). We believe this last step improves test accuracy since the decoder creates some distributional shift between the synthetic embeddings. Encoding and decoding the test set ensures the same shift is applied to the test set.

Before training, we again restrict the synthetic training set to the same size as the original non-synth training set for fair comparison against other private synthetic data baselines.

[7] https://pytorch.org/blog/how-to-train-state-of-the-art-models-using-torchvision-latest-primitives/

D Deferred Preliminaries

Definition D.1 (Gaussian Mixture Model (GMM); see e.g. [Rey+09])**.** *A Gaussian Mixture Model (GMM) is a parametric probability density function represented as a weighted sum of Gaussian component densities* $\sum_{i=1}^{k} w_i \cdot \mathcal{N}(\mu_i, \Sigma_i)$ *where* $w \geq 0$ *satisfies* $\sum_i w_i = 1$

We also require our loss function to be well-behaved so that small perturbations in the input space do not result in large perturbations in the label space. Lipschitz continuity and its generalization to Hölder continuity are standard assumptions for this purpose (see e.g., Axiotis, Cohen-Addad, Henzinger, et al. [ACH+24]).

Definition D.2 (Hölder continuity)**.** *We say a function* $f : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ *is* (z, λ)*-Hölder continuous if for all* $(\mathbf{x}, y), (\mathbf{x}', y) \in \mathcal{X} \times \mathcal{Y}$, $|f(\mathbf{x}, y) - f(\mathbf{x}', y)| \leq \lambda \|\mathbf{x} - \mathbf{x}'\|_2^z$.

D.1 Differential Privacy

We first recall the following preliminaries from differential privacy.

Definition D.3 (Differential privacy; Dwork, McSherry, Nissim, et al. [DMN+06])**.** *Given* $\varepsilon > 0$ *and* $\delta \in (0, 1)$*, a randomized algorithm* $\mathcal{A} : \mathcal{X}^n \to \mathcal{R}$ *is* (ε, δ)*-differentially private if, for every pair of neighboring datasets* $D, D' \in \mathcal{X}^n$ *that differ by a single entry and for all subsets* $U \subseteq \mathcal{R}$ *of the output space* $\mathcal{R}$*,*

$$\mathbf{Pr}\left[\mathcal{A}(D) \in U\right] \leq e^{\varepsilon} \cdot \mathbf{Pr}\left[\mathcal{A}(D') \in U\right] + \delta\,.$$

Definition D.4 (Laplace distribution)**.** *A random variable* x *follows the* Laplace distribution *with mean* μ *and scale parameter* $b > 0$*, denoted as* $x \sim \mathrm{Lap}(\mu, b)$*, if its probability density function is given by* $\frac{1}{2b} \exp\left(-\frac{|x-\mu|}{b}\right)$*. We use* $x \sim \mathrm{Lap}(b)$ *to denote* $x \sim \mathrm{Lap}(0, b)$*.*

A common method to ensure differential privacy involves adding Laplacian noise, with scale parameter proportional to the following notion:

Definition D.5 (L_1 sensitivity)**.** *Let* $x \sim y$ *denote neighboring databases that differ by a single entry. The* ℓ_1 *sensitivity of a function* f *is defined by*

$$\Delta_f = \max_{x,y:x\sim y} \|f(x) - f(y)\|_1.$$

Informally, the L_1 sensitivity of a function is the largest amount that a single entry in a database can affect f.

Definition D.6 (Laplace mechanism)**.** *Given a function* f*, an input* x*, and a privacy parameter* $\varepsilon > 0$*, the Laplace mechanism outputs* $f(x) + \eta$*, where* $\eta \sim \mathrm{Lap}\left(\frac{\Delta_f}{\varepsilon}\right)$*.*

The Laplace mechanism is a fundamental technique for establishing differential privacy:

Theorem D.7 (Dwork and Roth [DR14])**.** *The Laplace mechanism preserves* $(\varepsilon, 0)$*-differential privacy.*

An important property of differential privacy is that performing computation on a privatized dataset cannot lose additional privacy:

Theorem D.8 (Post-processing of differential privacy; Dwork and Roth [DR14])**.** *Let $\mathcal{M}$ be an (ε, δ)-differential private mechanism and g be any arbitrary random mapping. Then $g(\mathcal{M}(\cdot))$ is (ε, δ)-differentially private.*

Moreover, multiple computations on a dataset incur privacy cost in a natural manner.

Theorem D.9 (Basic composition of differential privacy; Dwork and Roth [DR14])**.** *Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be $(\varepsilon_1, \delta_1)$ and $(\varepsilon_2, \delta_2)$-DP mechanism, respectively. Then the composition $(\mathcal{M}_1(\cdot), \mathcal{M}_2(\cdot))$ is $(\varepsilon_1 + \varepsilon_2, \delta_1 + \delta_2)$-differentially private.*

On the other hand, executing a private mechanism on disjoint partitions of the same dataset does not incur any additional privacy cost.

Theorem D.10 (Parallel composition of differential privacy; McSherry [McS09])**.** *Let $\mathcal{M}$ be an (ε, δ)-DP mechanism and $D_1, \ldots, D_k$ k-disjoint subsets of the dataset D. Then the mechanism that outputs $(\mathcal{M}(D_1), \ldots, \mathcal{M}(D_k))$ is (ε, δ)-DP.*

There are more sophisticated composition results, but Theorem D.9 and Theorem D.10 suffice for our purposes.

D.2 Concentration Inequalities

Theorem D.11 (Hanson and Wright [HW71])**.** *Suppose $Y_i \sim_{i.i.d.} \mathcal{N}(\mu, \Sigma)$ for $i \in [N]$. Then with probability $1 - \beta$,*

$$\left\| \frac{1}{N} \sum_i Y_i - \mu \right\|_2 \leq \sqrt{\frac{\mathrm{Tr}(\Sigma)}{N}} + \sqrt{\frac{2 \|\Sigma\|_2 \log(1/\beta)}{N}} .$$

Theorem D.12 (Remark 5.40 in Vershynin [Ver10])**.** *Suppose $Y_i \sim_{i.i.d.} \mathcal{N}(\mu, \Sigma)$ for $i \in [N]$ and let $\bar{Y} = \frac{1}{N} \sum_{i \in [N]} Y_i$. Then with probability $1 - \beta$,*

$$\left\| \frac{1}{N} \sum_i (Y_i - \bar{Y})(Y_i - \bar{Y})^\top - \Sigma \right\|_2 \leq O\left(\sqrt{\frac{d}{N}} + \sqrt{\frac{\log(1/\beta)}{N}} \right) \|\Sigma\|_2 .$$

Theorem D.13 (Chernoff Bound; Corollary 4.6 in Mitzenmacher and Upfal [MU05])**.** *Let $X_1, X_2, \ldots, X_n \in \{0, 1\}$ be independent Bernoulli random variables and let $X = \sum_{i=1}^n X_i$. Then for any $\gamma \in (0, 1)$,*

$$\mathbf{Pr}\left[|X - \mathbb{E}[X]| \geq \gamma \cdot \mathbb{E}[X] \right] \leq 2 \exp\left(-\frac{\gamma^2 \cdot \mathbb{E}[X]}{3} \right) .$$

Theorem D.14 (Dvoretzky, Kiefer, and Wolfowitz [DKW56] and Massart [Mas90])**.** *Let $F(x)$ denote the CDF function for an arbitrary distribution. For $x_1, \ldots, x_n \sim_{i.i.d.} F$, write $F_n(x) = \frac{1}{n} \sum_{i=1}^n \mathbb{1}\{x = x_i\}$ to denote the n-sample empirical CDF function F_n. It holds that*

$$\Pr\left[\sup_x |F_n(x) - F(x)| > \gamma \right] \leq 2 \exp(-2n\gamma^2) .$$

E Private Gaussian Estimation

The first sample-optimal DP Gaussian mean/covariance estimation algorithms were due to Aden-Ali, Ashtiani, and Kamath [AAK21]. However, their algorithms require exponential running time. Recent transformations from robust algorithms to private algorithms obtained the same optimal sample complexities for mean estimation [HKM22] and covariance estimation [HKM+23] in polynomial time. We state their results below. See Hopkins, Kamath, Majid, et al. [HKM+23, Table 1, Table 2] for a detailed summary of prior algorithmic results. The sample complexities are tight up to logarithmic factors [KV18; KMS22; Nar24; PH24].

Theorem E.1 (Theorem 1.4 in Hopkins, Kamath, Majid, et al. [HKM+23])**.** *Let $\varepsilon, \delta, \alpha, \beta \in (0,1)$. Suppose we are provided sample access to a Gaussian distribution $\mathcal{N}(\mu, \Sigma)$ with unknown parameters. There is an (ε, δ)-DP mean estimation algorithm which outputs estimates $\hat{\mu}, \hat{\Sigma}$ such that $\|\hat{\mu} - \mu\|_2, \left\|\hat{\Sigma} - \Sigma\right\|_F \leq \alpha$ with probability $1 - \beta$. Moreover, the algorithm has sample complexity*

$$n = \tilde{O}\left(\frac{d^2 + \log^2(1/\beta)}{\alpha^2} + \frac{d^2 + \log(1/\beta)}{\alpha\varepsilon} + \frac{\log(1/\delta)}{\varepsilon}\right)$$

and $\mathrm{poly}(n)$ *running time.*

For the optimal sample complexity guarantees, we use Theorem E.1 to instantiate our `DP-Mean` and `DP-Covariance` subroutines. However, we note that any (ε, δ)-DP Gaussian estimation algorithm suffices. Indeed, by augmenting the assumptions with some (weak) priors about the mean and covariances, there are near-linear time estimators achieving nearly the same sample complexities. We state one example of such an estimator below.

Theorem E.2 (Theorems 3.1, 3.3 in Biswas, Dong, Kamath, et al. [BDK+20])**.** *Let $\varepsilon, \delta, \alpha, \beta \in (0,1)$. Suppose we are provided sample access to a Gaussian distribution $\mathcal{N}(\mu, \Sigma)$ with unknown parameters but are provided bounds $R, \lambda, \Lambda > 0$ such that $\|\mu\|_2 \leq R$ and $\lambda I \preceq \Sigma \preceq \Lambda I$. There is an (ε, δ)-DP mean estimation algorithm that outputs estimates $\hat{\mu}, \hat{\Sigma}$ such that $\|\hat{\mu} - \mu\|_2, \left\|\hat{\Sigma} - \Sigma\right\|_F \leq \alpha$ with probability $1 - \beta$. Moreover, we have:*

(i) For a general covariance matrix Σ, the algorithm has sample complexity

$$n = \tilde{O}\left(\left(\frac{d^2}{\alpha^2} + \frac{d^2\sqrt{\log(1/\delta)}}{\alpha\varepsilon} + \frac{d^{1.5}\sqrt{\log(1/\delta)\log(R + \Lambda/\lambda)}}{\varepsilon}\right)\log(1/\beta)\right)$$

and $\tilde{O}(nd^2)$ running time.

(ii) For a diagonal covariance matrix Σ, the algorithm has sample complexity

$$n = \tilde{O}\left(\left(\frac{d}{\alpha^2} + \frac{d\sqrt{\log(1/\delta)}}{\alpha\varepsilon} + \frac{d\sqrt{\log(1/\delta)\log(R + \Lambda/\lambda)}}{\varepsilon}\right)\log(1/\beta)\right)$$

and $\tilde{O}(nd)$ running time.

F Wasserstein Distance between GMMs

Suppose we have a GMM $\hat{\mathcal{D}} = \sum_i \hat{w}_i \cdot \mathcal{N}(\hat{\mu}_i, \hat{\Sigma}_i)$ and would like to understand its p-th order Wasserstein distance to $\mathcal{D}_{\text{GMM}}$. We will prove the following theorem.

Theorem F.1. *Let $\alpha, \gamma \in (0,1)$ and $R, \sigma > 0$. When $\|\hat{\mu}_i - \mu_i\|_2, \left\|\hat{\Sigma}_i - \Sigma_i\right\|_2 \le \alpha$, $\|\hat{w} - w\|_1 \le \gamma$, $\max_{i \neq j} \|\mu_i - \mu_j\| \le R$, and $\Sigma_i \preceq \sigma^2 I$, we have for any $z \in [1,2]$,*

$$W_z^z(\mathcal{D}_{\text{GMM}}, \hat{\mathcal{D}}) = O(\gamma R^z + \gamma d^{\frac{z}{2}} \sigma^z + \alpha^z + d^{\frac{z}{4}} \alpha^{\frac{z}{2}}).$$

We first consider the case where the number of components $k = 1$.

F.1 One-Component Case

For two Gaussian distributions $\mathcal{N}(\mu, \Sigma), \mathcal{N}(\tilde{\mu}, \tilde{\Sigma})$, it is known that

$$W_2^2(\mathcal{N}(\mu, \Sigma), \mathcal{N}(\tilde{\mu}, \tilde{\Sigma})) = \|\mu - \tilde{\mu}\|_2^2 + \left\|\Sigma^{\frac{1}{2}} - \tilde{\Sigma}^{\frac{1}{2}}\right\|_F^2.$$

We can translate the second term to a bound on the covariance matrices

$$W_2^2(\mathcal{N}(\mu, \Sigma), \mathcal{N}(\tilde{\mu}, \tilde{\Sigma})) \le \|\mu - \tilde{\mu}\|_2^2 + \sqrt{d} \left\|\Sigma - \tilde{\Sigma}\right\|_F,$$

since the Powers–Størmer inequality implies $\left\|\Sigma^{\frac{1}{2}} - \tilde{\Sigma}^{\frac{1}{2}}\right\|_F^2 \le \text{Tr}(|\Sigma - \tilde{\Sigma}|)$ and the standard trace-norm inequality gives $\text{Tr}(|\Sigma - \tilde{\Sigma}|) \le \sqrt{d} \left\|\Sigma - \tilde{\Sigma}\right\|_F$. Finally, for any $z \in [1,2]$, a standard concavity split yields

$$W_z^z(\mathcal{N}(\mu, \Sigma), \mathcal{N}(\tilde{\mu}, \tilde{\Sigma})) \le W_2^z(\mathcal{N}(\mu, \Sigma), \mathcal{N}(\tilde{\mu}, \tilde{\Sigma})) \le 2^{\frac{z}{2}-1} \|\mu - \tilde{\mu}\|_2^z + 2^{\frac{z}{2}-1} d^{\frac{z}{4}} \left\|\Sigma - \tilde{\Sigma}\right\|_F^{\frac{z}{2}}.$$

F.2 Multi-Component Case

Suppose now that $k \ge 2$ and that up to permutation, we estimated the means and covariances in Euclidean and spectral norm, respectively. Thus $\|\hat{\mu}_i - \mu\|_2, \left\|\hat{\Sigma}_i - \Sigma_i\right\| \le \alpha$. Moreover, suppose that the weights have been estimated in total variation distance. That is, $\|\hat{\mathbf{w}} - \mathbf{w}\|_1 \le \gamma$. We can analyze

$$W_z^z(\mathcal{D}_{\text{GMM}}, \hat{\mathcal{D}}) \le 2^{z-1} W_z^z(\mathcal{D}_{\text{GMM}}, \mathcal{D}_w) + 2^{z-1} W_z^z(\mathcal{D}_w, \hat{\mathcal{D}}).$$

Here $\mathcal{D}_w$ is the auxiliary GMM whose components are equal to that of $\mathcal{D}_{\text{GMM}}$ but the weights are the estimated weights. Then, we need only bound separates cases when the parameters differ and when the weights differ.

We note that by the one-component case,

$$W_z^z(\mathcal{D}_w, \hat{\mathcal{D}}) \le \sum_{i=1}^k \hat{w}_i W_z^z(\mathcal{N}(\mu_i, \Sigma_i), \mathcal{N}(\hat{\mu}_i, \Sigma_i)) \le \max_i 2^{\frac{z}{2}-1} \|\mu - \tilde{\mu}\|_2^z + 2^{\frac{z}{2}-1} d^{\frac{z}{4}} \left\|\Sigma - \tilde{\Sigma}\right\|_F^{\frac{z}{2}}.$$

Now we consider the case when only the weights differ in ℓ_1-norm by at most γ. Under the optimal coupling in total variation distance of the weights, the transportation cost is 0 when the

random vectors coincide and when they differ, we can transport the mass from the component to an arbitrary component with total cost at most

$$\gamma \cdot \max_{i \neq j} 2^{\frac{z}{2}-1} \|\mu_i - \mu_j\|_2^z + 2^{\frac{z}{2}-1} \left\|\Sigma_i^{\frac{1}{2}} - \Sigma_j^{\frac{1}{2}}\right\|_F^z .$$

All in all, when $\|\hat{\mu}_i - \mu_i\|_2, \left\|\hat{\Sigma}_i - \Sigma_i\right\|_F \leq \alpha$, $\|\hat{w} - w\|_1 \leq \gamma$, $\max_{i \neq j} \|\mu_i - \mu_j\| \leq R$, and $\Sigma_i \preceq \sigma^2 I$, we have for any $z \in [1, 2]$ that

$$\begin{aligned} W_z^z(\mathcal{D}_{\mathrm{GMM}}, \hat{\mathcal{D}}) &\leq 2^{\frac{3z}{2}-2}\gamma R^z + 2^{\frac{5z}{2}-2}\gamma d^{\frac{z}{2}}\sigma^z + 2^{\frac{3z}{2}-2}\alpha^z + 2^{\frac{3z}{2}-2} d^{\frac{z}{4}}\alpha^{\frac{z}{2}} \\ &= O(\gamma R^z + \gamma d^{\frac{z}{2}}\sigma^z + \alpha^z + d^{\frac{z}{4}}\alpha^{\frac{z}{2}}) . \end{aligned}$$

G Theoretical Analysis

We now analyze the privacy, scalability, and utility guarantees of Algorithm 1.

G.1 Privacy Analysis

In this section, we formally prove that Algorithm 1 is differentially private.

Theorem G.1. *Algorithm 1 is (ε, δ)-DP.*

Proof. We can view Algorithm 1 as a composition of 5 subroutines: `DP-Cluster`, `DP-Mean`, `DP-Covariance`, `DP-FilterEmbedding`, `DP-FilterImage`, each of which are $(\varepsilon/5, \delta/5)$-DP. By simple composition (Theorem D.9), Algorithm 1 satisfies (ε, δ)-DP. □

While the analysis above was for unconditional generation, the result extends immediately to conditional generation. Indeed, differential privacy satisfies parallel composition (c.f. Theorem D.10), which means there is no additional privacy loss incurred by running differentially private algorithms on separate, non-overlapping parts of the data. Hence there is no additional privacy loss compared to running the pipeline over each of the classes in parallel.

G.2 Scalability Analysis

Suppose $T_{\texttt{Encode}}$ is the runtime required to apply a fixed embedding to each input image, $T_{\texttt{Decode}}$ is the runtime required to decode a fixed embedding, and $T_{\texttt{DP-FilterEmbedding}}(n), T_{\texttt{DP-FilterImage}}(n)$ are the runtimes of the (possibly private) filtering subroutines for embeddings and images, respectively. We have the following running time guarantee of Algorithm 1.

Theorem G.2. *For general covariance structure GMMs, Algorithm 1 can be implemented in time*

$$\tilde{O}\left(nT_{\texttt{Encode}} + nd^2 + T_{\texttt{DP-FilterEmbedding}}(n) + nT_{\texttt{Decode}} + T_{\texttt{DP-FilterImage}}(n)\right).$$

while for diagonal covariance structure GMMs, Algorithm 1 can be implemented in time

$$\tilde{O}\left(nT_{\texttt{Encode}} + nd + T_{\texttt{DP-FilterEmbedding}}(n) + nT_{\texttt{Decode}} + T_{\texttt{DP-FilterImage}}(n)\right).$$

Proof. The only subroutines that have not been accounted for is the running times for `DP-Cluster`, `DP-Mean`, and `DP-Covariance`. There are implementations of private clustering algorithms with $\tilde{O}(nd)$ running time [CEL+22; CEM+22]. Also, there are implementations of private Gaussian estimation algorithms with $\tilde{O}(nd^2)$ or $\tilde{O}(nd)$ running times, respectively, for general and diagonal covariance structure GMMs (Theorem E.2). □

G.3 Learning GMMs via k-Means Clustering

In this section, we demonstrate that our clustering-based DP GMM algorithm can recover separated GMMs. Our analysis is inspired by the non-private algorithm of Awasthi and Vijayaraghavan [AV18].

We begin by arguing that the number of sample from each mixture concentrates about its mean.

Lemma G.3. *Let $\gamma \in (0,1)$ and let $\mathcal{D}_{\mathrm{GMM}} = \sum_{i=1}^{k} w_i \mathcal{N}(\mu_i, \Sigma_i)$ be a Gaussian mixture.*

(i) If the number of samples is at least $N = \Omega\left(\gamma^{-2}\log(1/\beta)\right)$, then the number of samples N_i drawn from the i-th component satisfies $\sup_i |N_i/N - w_i| \le \gamma$ with probability $1-\beta$.

(ii) If $N = \Omega\left(\frac{\log(k/\beta)}{\gamma^2 w_{\min}}\right)$, then $|N_i - w_i N| \le \gamma w_i N$ with probability $1-\beta$.

Proof. The proof of (i) is an straightforward application of the DKW inequality (Theorem D.14), where we view the mixture components as a discrete one-dimensional distribution. Similarly, we can prove (ii) by applying a Chernoff bound (Theorem D.13) where we view drawing a sample from the i-th mixture component as a Bernoulli outcome and apply a union bound over all components. □

Next, we provide a high-probability upper bound on the optimal k-means clustering cost. This will soon be useful when arguing about the behavior of an approximate solution.

Lemma G.4. *Let $\mathcal{D}_{\mathrm{GMM}} = \sum_{i=1}^{k} w_i \mathcal{N}(\mu_i, \Sigma_i)$ be a Gaussian mixture for $\Sigma_i \preceq \sigma^2 I$. Let $A \in \mathbb{R}^{N\times d}$ denote the matrix of data points and $M^\star \in \{\mu_1, \dots, \mu_k\}^{N\times d}$ is the matrix obtained from A by replacing each row with the mean of the component that generated it. Suppose the number of samples is at least $N = \Omega\left(\frac{1}{w_{\min}}\log(k/\delta)\right)$. Then with probability $1-\delta$,*

$$\|A - M^\star\|_2^2 \le \frac{4}{3}\sigma^2 N\,.$$

Proof. The proof will be via an application of sample covariance concentration (Theorem D.12). We partition the samples $C_1 \cup \cdots \cup C_k$ by the components that generated them. Note that $|C_i| = N_i$. Say μ_i is the mean of the Gaussian component that generated the points in C_i.

We claim that it suffices to show that

$$\left\| \frac{1}{N_i} \sum_{x\in C_i} (x-\mu_i)(x-\mu_i)^\top \right\|_2^2 = \max_{v\in\mathbb{R}^d, \|v\|_2=1} \frac{1}{N_i}\sum_{x\in C_i} \langle v, x-\mu_i\rangle^2 \le \frac{4}{3}\sigma^2$$

for each component $i \in [k]$. To see this, observe that

$$\|A - M^\star\|_2^2 = \max_{v\in\mathbb{R}^d, \|v\|_2=1} v^\top \left(\sum_{i=1}^{k} \sum_{x\in C_i} (x-\mu_i)(x-\mu_i)^\top \right) v = \max_{v\in\mathbb{R}^d, \|v\|_2=1} \sum_{i=1}^{k}\sum_{x\in C_i} \langle v, x-\mu_i\rangle^2\,.$$

Proving the claim implies a bound on the last expression, which would then conclude the proof.

Now, to see the claim, we first apply Lemma G.3 with $\gamma = 1/2$ to see that $N_i \ge \frac{1}{2} w_i N$ with probability $1-\beta$. Then, conditional on this event, we can apply Theorem D.12 with appropriate constants to see that $N_i \ge \frac{1}{2} w_i N$ is sufficiently large to conclude the proof. □

We say an algorithm is a (ζ, η)-approximate k-means algorithm if produces a set of k centers that induces a clustering cost of at most $\zeta \cdot \mathsf{OPT} + \eta$ where OPT is the optimal k-means clustering cost.

Lemma G.5. *Let $\mathcal{D}_{\mathrm{GMM}} = \sum_{i=1}^{k} w_i \mathcal{N}(\mu_i, \Sigma_i)$ be a Gaussian mixture for $\Sigma_i \preceq \sigma^2 I$. Then, if the number of samples is at least $N = \Omega\left(\frac{d+\log(k/\delta)}{w_{\min}}\right)$, any (ζ, η)-approximate k-means algorithm for $\eta = o(\gamma \sigma^2 dN)$ outputs k centers $\nu_1, \ldots, \nu_k$ such that*

$$\max_i \min_j \|\mu_i - \nu_j\|_2^2 \leq \frac{12\zeta}{w_{\min}} \sigma^2 d$$

with probability $1 - \delta$.

Proof. Let $A \in \mathbb{R}^{N \times d}$ be the matrix whose rows consists of sample points from $\mathcal{D}_{\mathrm{GMM}}$ and $M^\star \in \mathbb{R}^{N \times d}$ be the centers of the Gaussians that generated the corresponding point. We see that the optimal k-means clustering has cost at most $\|A - M^\star\|_F^2$. By Lemma G.4, we can upper bound $\|A - M^\star\|_F^2 \leq \frac{4}{3}\sigma^2 dN$.

Now, suppose towards a contradiction that there is some μ_i such that for every center ν_j output by the k-means approximation algorithm, $\|\mu_i - \nu_j\|_2^2 \geq \frac{12\zeta}{w_{\min}}\sigma^2 d$. Consider the cost paid by the points C_i generated from the i-th component. We have by generalized triangle inequality,

$$\sum_{x \in C_i} \|x - \nu(x)\|_2^2 \geq \sum_{x \in C_i} \left[\frac{1}{2}\|\mu_i - \nu(x)\|_2^2 - \|x - \mu_i\|_2^2\right].$$

By assumption, the first term, i.e., $\frac{1}{2}\|\mu_i - \nu(x)\|_2^2$, contributes at least $\frac{1}{2}|C_i| \cdot \frac{12\zeta}{w_{\min}}\sigma^2 d$. We can further lower bound this by $\frac{1}{4} w_i N \cdot \frac{12\zeta}{w_{\min}}\sigma^2 d \geq 3\zeta\sigma^2 dN$ using a Chernoff bound (Lemma G.3). On the other hand, the second term can be (loosely) upper bounded by $\|A - M^\star\|_F^2 \leq \frac{4}{3}\sigma^2 dN$. But then for sufficiently large N, this is at least $\frac{4}{3}\zeta\sigma^2 dN + \eta$, which contradicts the approximation guarantee. □

Lemma G.5 essentially ensures that running a k-means algorithm recovers the means of the Gaussian mixture model up to some error that is independent of the number of points. Then, assuming the means are well-separated, we can correctly classify all points of each component of the mixture model. Estimating the mean and covariance within each class then recovers the underlying mean and covariance.

Theorem G.6. *Let $\mathcal{D}_{\mathrm{GMM}} = \sum_{i=1}^{k} w_i \mathcal{N}(\mu_i, \Sigma_i)$ be a Gaussian mixture for $\Sigma_i \preceq \sigma^2 I$. Suppose the number of samples is at least $N = \Omega\left(\frac{d+\log(k/\beta)}{w_{\min}\alpha^2}\right)$ and let $\nu_1, \ldots, \nu_k$ be the output centers of some (ζ, η)-approximate k-means algorithm for $\eta = o(\zeta\sigma^2 dN)$. Let $C_1, \ldots, C_k$ denote the partition of sample points induced by the centers. If*

$$\Delta := \min_{i \neq j} \|\mu_i - \mu_j\|_2 \geq 3\sigma\left[\sqrt{d} + \sqrt{2\log(3N/\delta)} + \sqrt{\frac{12\zeta d}{w_{\min}}}\right],$$

then with probability $1 - \beta$, for every $i \in [k]$,

(i) there is a unique center $\nu_{j(i)} = \operatorname{argmin}_j \|\nu_j - \mu_i\|_2$ that is closest to μ_i.

(ii) Furthermore, each $C_{j(i)}$ only contains points sampled from the i-th component $\mathcal{N}(\mu_i, \Sigma_i)$.

(iii) $|C_{j(i)}| \geq \frac{1}{\alpha^2} w_i N$ for all $i \in [k]$.

Proof. Let $\mu(x), \Sigma(x)$ denote the parameters of the component Gaussian that generated the sample x and N_i denote the number of samples that was generated from the i-th component. We condition on the following events, each of which occurs with probability $1 - \delta/3$:

$$\|\nu_i - \mu_i\|_2 \leq \sqrt{\frac{12\zeta\sigma^2 d}{w_{\min}}}, \qquad \forall i \in [k]\,, \qquad \text{(by Lemma G.5)}$$

$$\|x - \mu(x)\|_2 \leq \sigma\left[\sqrt{d} + \sqrt{2\log(3N/\delta)}\right], \qquad \forall x\,, \qquad \text{(by Theorem D.11)}$$

$$N_i \geq \frac{1}{2} w_i N, \qquad \forall i \in [k]\,. \qquad \text{(by Theorem D.13)}$$

Let $\mathcal{E}$ denote the intersection of all events above. Henceforth, we always condition on $\mathcal{E}$ occurring.

(i): Under a slight abuse of notation, we relabel ν_i to be the closest output center to μ_i and suppose towards a contradiction that we have $\nu_i = \nu_j$ for $i \neq j$. By a reverse triangle inequality,

$$\begin{aligned}
\|\mu_i - \nu_j\|_2 &\geq \|\mu_i - \mu_j\|_2 - \|\nu_j - \mu_j\|_2 \\
&\geq \Delta - \sqrt{\frac{12\zeta\sigma^2 d}{w_{\min}}} && \text{(conditioned on } \mathcal{E}) \\
&\geq 2\sqrt{\frac{12\zeta\sigma^2 d}{w_{\min}}} \\
&> \|\mu_i - \nu_i\|_2\,.
\end{aligned}$$

We can thus proceed using the relabelled notation for the set of centers as it is well-defined.

(ii): We now claim that the partition $C_1, \ldots, C_k$ correctly classifies all points. Specifically, if a point x was sampled from the i-th component $\mathcal{N}(\mu_i, \Sigma_i)$, then $x \in C_i$.

Let $\nu(x)$ denote the unique closest center to $\mu(x)$. Fix a sample x and let $\mu(x) \neq \mu \in \{\mu_1, \ldots, \mu_k\}$ and $\nu(x) \neq \nu \in \{\nu_1, \ldots, \nu_k\}$. We have

$$\begin{aligned}
\|\nu(x) - x\|_2 &\leq \|\nu(x) - \mu(x)\|_2 + \|x - \mu(x)\|_2 \\
&\leq \sqrt{\frac{12\zeta\sigma^2 d}{w_{\min}}} + \sigma[\sqrt{d} + \sqrt{2\log(N/\delta)}]\,. && \text{(conditioned on } \mathcal{E})
\end{aligned}$$

Again by a reverse triangle inequality, we have

$$\begin{aligned}
\|x - \nu\|_2 &\geq \|\mu(x) - \mu\|_2 - \|\mu(x) - x\|_2 - \|\nu - \mu\|_2 \\
&\geq \Delta - \sigma[\sqrt{d} + \sqrt{2\log(N/\delta)}] - \sqrt{\frac{12\zeta\sigma^2 d}{w_{\min}}} && \text{(conditioned on } \mathcal{E}) \\
&> \|\nu(x) - x\|_2\,. && \text{(by calculation above)}
\end{aligned}$$

This shows that each C_i only contains i.i.d. samples from $\mathcal{N}(\mu_i, \Sigma_i)$ with high probability.

(iii): This follows by Lemma G.3.

□

Theorem G.6 essentially shows that k-means clustering with an additional Lloyd step will recover the true means of the mixture model. Similarly, the intra-cluster sample covariance will recover the true covariances of the mixture model.

Theorem G.7. *Let $\mathcal{D}_{\mathrm{GMM}} = \sum_{i=1}^{k} w_i \mathcal{N}(\mu_i, \Sigma_i)$ be a Gaussian mixture for $\Sigma_i \preceq \sigma^2 I$. Suppose we are given N samples from $\mathcal{D}_{\mathrm{GMM}}$ such that*

$$N = \tilde{\Omega}\left(\frac{d + \log(k/\beta)}{w_{\min}} + \frac{k^2 \log(1/\beta)}{\gamma^2} + \frac{k \log(k/\beta)}{\varepsilon\gamma} + \frac{d^2 + \log^2(1/\beta)}{w_{\min}\alpha^2} + \frac{d^2 + \log(1/\beta)}{w_{\min}\alpha\varepsilon} + \frac{\log(1/\delta)}{w_{\min}\varepsilon}\right)$$

and also

$$\Delta \coloneqq \min_{i \neq j} \|\mu_i - \mu_j\|_2 \geq 3\sigma\left[\sqrt{d} + \sqrt{2\log(3N/\beta)} + \sqrt{\frac{12\zeta d}{w_{\min}}}\right].$$

Then instantiating Algorithm 1 with

(1) `DP-Cluster` *as an (ε, δ)-DP (ζ, η)-approximate k-means algorithm for $\eta = o(\zeta\sigma^2 dN)$,*

(2) `DP-Mean`, `DP-Covariance` *as the (ε, δ)-DP Gaussian estimation algorithm from Theorem E.1*

yields an (ε, δ)-DP algorithm that outputs with probability $1 - \beta$:

(i) Weight estimates $\hat{w}_i$ such that $\|\hat{w} - w\|_1 \leq \gamma$.

(ii) Mean estimates $\hat{\mu}_i$ such that $\|\hat{\mu}_i - \mu\|_2 \leq \alpha$.

(iii) Covariance estimates $\hat{\Sigma}_i$ such that $\left\|\hat{\Sigma}_i - \Sigma_i\right\|_F \leq \alpha$.

Proof. The proof follows by augmenting `DP-Cluster` with a simple Laplace mechanism (Theorem D.7) to privately estimate the weights and applying Theorem G.6 and Theorem E.1. □

We note that while non-private clustering-based GMM algorithms (see Section 1.3) with weaker separation conditions exist, they require specific clustering algorithms. In comparison, we need only black-box access to an approximate k-means clustering algorithm.

G.4 Utility Analysis

Now we translate GMM parameter estimates to distributional estimates, which implies that minimizing the objective function over the estimated distribution will also approximately minimize the objective function over the true distribution.

G.4.1 Learning in Total Variation Distance

We begin with learning a GMM in total variation distance. When translating this to an approximation in the objective function, we then require a bound on the maximum absolute function value.

Lemma G.8. *Let $Z = (X, Y)$ be a joint feature-label distribution for $Y \in [c]$ where each conditional distribution $(X \mid Y = y) \sim \mathcal{D}^{(y)}$. Suppose the distribution $\tilde{Z} = (\tilde{X}, Y)$ has conditional distributions $(\tilde{X} \mid Y = y) \sim \widehat{\mathcal{D}}^{(y)}$ satisfying $\mathrm{TV}(\mathcal{D}^{(y)}, \widehat{\mathcal{D}}^{(y)}) \leq \alpha$ for all $y \in [c]$. Then for any function f, we have*

$$\mathop{\mathbb{E}}_{Z}[f(Z)] \leq \mathop{\mathbb{E}}_{\tilde{Z}}\left[f(\tilde{Z})\right] + \alpha \cdot \max|f|.$$

Proof. Fix any $y \in [c]$. By the definition of total variation distance, for each y there exists a coupling of $D(y)$ and $\tilde{D}(y)$ such that if $(X, \tilde{X})$ are drawn from this coupling, then

$$\mathbf{Pr}\left[X \neq \tilde{X} \mid Y = y\right] \leq \mathrm{TV}(D(y), \tilde{D}(y)) \leq \alpha.$$

Let us construct the following joint distribution over $(X, \tilde{X}, Y)$:

- First draw $Y \sim \mathcal{D}_Y$, the marginal distribution of Y in both Z and $\tilde{Z}$.
- Given $Y = y$, draw $(X, \tilde{X})$ from a coupling of $D(y)$ and $\tilde{D}(y)$ that achieves total variation distance at most α and such that $\mathbf{Pr}\left[X \neq \tilde{X} \mid Y = y\right] \leq \alpha$.

Define the event $\mathcal{E}$ to be the event that $X \neq \tilde{X}$. By the law of total probability and the bound on total variation,

$$\mathbf{Pr}\left[\mathcal{E}\right] = \mathop{\mathbb{E}}_{Y}\left[\mathbf{Pr}\left[X \neq \tilde{X} \mid Y\right]\right] \leq \alpha.$$

Now consider evaluating f under $Z = (X, Y)$ and under $\tilde{Z} = (\tilde{X}, Y)$. We decompose the expectation:

$$\mathbb{E}\left[f(X, Y)\right] = \mathbb{E}\left[f(X, Y) \mid \neg\mathcal{E}\right] \cdot \mathbf{Pr}\left[\neg\mathcal{E}\right] + \mathbb{E}\left[f(X, Y) \mid \mathcal{E}\right] \cdot \mathbf{Pr}\left[\mathcal{E}\right].$$

On the event $\neg\mathcal{E}$, we have $X = \tilde{X}$, so $f(X, Y) = f(\tilde{X}, Y)$. Hence,

$$\mathbb{E}\left[f(X, Y)\right] = \mathbb{E}\left[f(\tilde{X}, Y) \mid \mathcal{E}\right] \cdot \mathbf{Pr}\left[\mathcal{E}\right] + \mathbb{E}\left[f(X, Y) \mid \mathcal{E}\right] \cdot \mathbf{Pr}\left[\mathcal{E}\right].$$

We now upper bound the second term:

$$\mathbb{E}\left[f(X, Y) \mid \mathcal{E}\right] \cdot \mathbf{Pr}\left[\mathcal{E}\right] \leq \max|f| \cdot \mathbf{Pr}\left[\mathcal{E}\right] \leq \alpha \cdot \max|f|.$$

Since f is a loss function and thus non-negative, then we have

$$\begin{aligned}\mathbb{E}\left[f(X, Y)\right] &\leq \mathbb{E}\left[f(\tilde{X}, Y) \mid \mathcal{E}\right] \cdot \mathbf{Pr}\left[\mathcal{E}\right] + \alpha \cdot \max|f| \\ &\leq \mathbb{E}\left[f(\tilde{X}, Y)\right] + \alpha \cdot \max|f| = \mathbb{E}\left[f(\tilde{Z})\right] + \alpha \cdot \max|f|.\end{aligned}$$

This completes the proof. □

Theorem G.9. *Let $\varepsilon, \delta, \alpha, \beta \in (0, 1)$ and f be a loss function. Let $Z = (X, Y)$ is a joint feature-label distribution for $Y \in [c]$ where each conditional distribution $(X \mid Y = y) \sim \mathcal{D}^{(y)}_{\mathrm{GMM}} = \sum_{i=1}^{k} w_i^{(y)} \mathcal{N}(\mu_i^{(y)}, \Sigma_i^{(y)})$ follows a Gaussian mixture law for $\Sigma_i^{(y)} \preceq \sigma^2 I$. Suppose we are given $N^{(y)}$ samples from each conditional distribution $\mathcal{D}^{(y)}_{\mathrm{GMM}}$ such that*

$$N^{(y)} = \tilde{\Omega}\left(\frac{d + \log(k/\beta)}{w^{(y)}_{\min}} + \frac{k^2 \log(1/\beta)}{\alpha^2} + \frac{k \log(k/\beta)}{\varepsilon\alpha} + \frac{d^2 + \log^2(1/\beta)}{w^{(y)}_{\min}\alpha^2} + \frac{d^2 + \log(1/\beta)}{w^{(y)}_{\min}\alpha\varepsilon} + \frac{\log(1/\delta)}{w^{(y)}_{\min}\varepsilon}\right)$$

and also

$$\Delta^{(y)} := \min_{i \neq j} \left\|\mu_i^{(y)} - \mu_j^{(y)}\right\|_2 \geq 3\sigma\left[\sqrt{d} + \sqrt{2\log(3N/\beta)} + \sqrt{\frac{12\zeta d}{w^{(y)}_{\min}}}\right].$$

Then running Algorithm 1 on each class with

(1) `DP-Cluster` *as an* (ε, δ)*-DP* (ζ, η)*-approximate* k*-means algorithm for* $\eta = o(\zeta\sigma^2 dN)$*,*

(2) `DP-Mean`*,* `DP-Covariance` *as the* (ε, δ)*-DP Gaussian estimation algorithm from Theorem E.1*

yields an (ε, δ)*-DP algorithm that outputs a distribution* $\tilde{Z} = (\tilde{X}, Y)$ *such that*

$$\mathop{\mathbb{E}}_{Z}\left[f(Z)\right] \leq \mathop{\mathbb{E}}_{\tilde{Z}}\left[f(\tilde{Z})\right] + \alpha \cdot \max|f|\,.$$

with probability $1 - \beta$*.*

Proof. Since $\mathrm{TV}(\mathcal{N}(\mu, \Sigma), \mathcal{N}(\mu', \Sigma')) = O(\|\mu - \mu'\|_2 + \|\Sigma - \Sigma'\|_F)$, Theorem G.7 guarantees that we learn each conditional distribution within α-TV distance. An application of Lemma G.8 concludes the proof. □

G.4.2 Learning in Wasserstein Distance

For potentially unbounded functions, we derive a second result, which requires learning the GMM in an appropriate Wasserstein distance. Under this measure of distributional distance, we require the objective function to be Hölder continuous.

Lemma G.10. *Let* $Z = (X, Y)$ *be a joint feature-label distribution for* $Y \in [c]$ *where each conditional distribution* $(X \mid Y = y) \sim \mathcal{D}^{(y)}$*. Suppose the distribution* $\tilde{Z} = (\tilde{X}, Y)$ *has conditional distributions* $(\tilde{X} \mid Y = y) \sim \widehat{\mathcal{D}}^{(y)}$ *satisfying* $W_z(\mathcal{D}^{(y)}, \widehat{\mathcal{D}}^{(y)}) \leq \alpha$ *for all* $y \in [c]$*. Then for any* (λ, z)*-Hölder continuous function* f*, we have*

$$\mathop{\mathbb{E}}_{Z}\left[f(Z)\right] \leq \mathop{\mathbb{E}}_{\tilde{Z}}\left[f(\tilde{Z})\right] + \lambda \cdot \alpha^z\,.$$

Proof. By definition,

$$f(x, y) \leq f(x', y) + \lambda \left\|x - x'\right\|_2^z$$

for any x, x', y. Then, taking the expectation of this inequality under the optimal W_z^z coupling yields the result. □

Theorem G.11. *Let* $\varepsilon, \delta, \alpha, \beta \in (0, 1)$ *and* f *be a* (λ, z)*-Hölder continuous loss function for* $z \in [1, 2]$*. Let* $Z = (X, Y)$ *is a joint feature-label distribution for* $Y \in [c]$ *where each conditional distribution* $(X \mid Y = y) \sim \mathcal{D}^{(y)}_{\mathrm{GMM}} = \sum_{i=1}^{k} \mathcal{N}(\mu_i^{(y)}, \Sigma_i^{(y)})$ *follows a Gaussian mixture law with* $\min_{i \neq j} \left\|\mu_i^{(y)} - \mu_j^{(y)}\right\|_2 \leq R$ *and* $\Sigma_i^{(y)} \preceq \sigma^2 I$*. Suppose we are given* $N^{(y)}$ *samples from each conditional distribution* $\mathcal{D}^{(y)}_{\mathrm{GMM}}$ *such that*

$$N^{(y)} = \tilde{\Omega}\left(\frac{d + \log(k/\beta)}{w_{\min}} + \frac{(R^{2z} + d^z\sigma^{2z})k^2\log(1/\beta)}{\alpha^2} + \frac{(R^z + d^{\frac{z}{2}}\sigma^z)k\log(k/\beta)}{\varepsilon\alpha} + \frac{d^3 + d\log^2(1/\beta)}{w_{\min}^{(y)}\alpha^{\frac{4}{z}}} + \frac{d^{\frac{5}{2}} + d^{\frac{1}{2}}\log(1/\beta)}{w_{\min}^{(y)}\alpha^{\frac{2}{z}}\varepsilon} + \frac{\log(1/\delta)}{w_{\min}^{(y)}\varepsilon}\right)$$

and also

$$\Delta^{(y)} := \min_{i \neq j} \left\|\mu_i^{(y)} - \mu_j^{(y)}\right\|_2 \geq 3\sigma\left[\sqrt{d} + \sqrt{2\log(3N/\beta)} + \sqrt{\frac{12\zeta d}{w_{\min}^{(y)}}}\right].$$

Then running Algorithm 1 on each class with

(1) `DP-Cluster` *as an* (ε, δ)*-DP* (ζ, η)*-approximate* k*-means algorithm for* $\eta = o(\zeta\sigma^2 dN)$*,*

(2) `DP-Mean`*,* `DP-Covariance` *as the* (ε, δ)*-DP Gaussian estimation algorithm from Theorem E.1*

yields an (ε, δ)*-DP algorithm that outputs a distribution* $\tilde{Z} = (\tilde{X}, Y)$ *such that*

$$\mathop{\mathbb{E}}_{Z}[f(Z)] \leq \mathop{\mathbb{E}}_{\tilde{Z}}\left[f(\tilde{Z})\right] + \lambda \cdot \alpha$$

with probability $1 - \beta$*.*

Proof. The sample complexity follows from adjusting the error parameters in Theorem G.7 to satisfy the W_z error bound from Theorem F.1

$$W_z^z(\mathcal{D}_{\mathrm{GMM}}, \hat{\mathcal{D}}) = O(\gamma R^z + \gamma d^{\frac{z}{2}}\sigma^z + \alpha^z + d^{\frac{z}{4}}\alpha^{\frac{z}{2}})\,.$$

Then, the function estimation guarantee follows from Lemma G.10. □